

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«На правах рукопису»
УДК _____

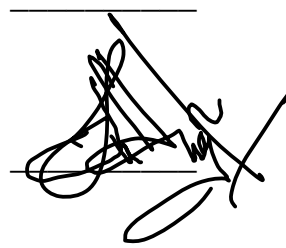
До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Система автоматичного паркування автомобілів на базі технології
інтернету речей»**

Виконав (-ла):
студент (-ка) VI курсу, групи ІА-92МПВ
Клімов Олександр Олександрович

Керівник:
Доцент, к.т.н., доц.
Писаренко Андрій Володимирович

Рецензент:
доц., каф. ТК, к.т.н., доц.
Ткач М.М.



Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Клімов Олександр Олександрович

1. Тема дисертації «Система автоматичного паркування автомобілів на базі технології інтернету речей», науковий керівник дисертації Писаренко Андрій Володимирович, к.т.н., доцент, затверджені наказом по університету від «26» 10 2020 р. №3132-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження паркінг

4. Вихідні дані система автоматичного паркування автомобілів з використанням локального позиціонування

5. Перелік завдань, які потрібно розробити 1) огляд і аналіз існуючих рішень; 2) технології локального позиціонування; 3) структурна схема системи; 4) функціональна схема системи; 5) Схема моделювання системи в MATLAB; 6) стартап-проект.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу 1) структурна схема системи; 2) функціональна схема системи; 3) схема моделі в MATLAB; 4) діаграма прецедентів системи; 5) діаграма послідовностей системи.

7. Орієнтовний перелік публікацій

8. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з предметною областю	09.09.2020	
2	Аналіз існуючих рішень	17.09.2020	
3	Ознайомлення з методами локального позиціонування	24.09.2020	
4	Розробка структурної схеми	28.09.2020	
5	Розробка функціональної схеми	02.10.2020	
6	Розробка моделі в MATLAB	09.10.2020	
7	Оформлення ПЗ	03.12.2020	

Студент

Олександр КЛІМОВ

Науковий керівник

Андрій ПИСАРЕНКО

РЕФЕРАТ

Магістерська дисертація освітньо-кваліфікаційного рівня «магістр» на тему «Система автоматичного паркування автомобілів на базі технології інтернету речей»: 115 с., 6 розділів, 27 таблиць, 30 рис., 25 джерело, 9 додатків.

З кожним роком кількість автомобілів на дорогах населених пунктів стрімко збільшується, тому питання парковки стає все більш проблемним в транспортній системі великих міст. Невпорядковані автостоянки є незручними як для клієнтів так і для адміністрації.

Метою дисертації є підвищення зручності користування паркінгами за рахунок впровадження системи автоматичного паркування.

В роботі було розроблено структуру системи, обрано окремі елементи, створено модель паркування в пакеті MATLAB. Розроблено функціональні схеми вузлів системи.

Ключові слова: автоматичне паркування, інтернет речей, IOT, MATLAB, самокеровані автомобілі.

ABSTRACT

Master's dissertation of the educational qualification level "Master" on the topic «IoT based automatic car parking system»: 115 p., 6 sections, 27 tables, 30 figures., 25 sources, 9 annexes.

The amount of traffic in cities increases, that's why the question of parking becomes more and more problematic in the transport system of big cities. Disordered parking is inconvenient as for clients and for administration.

The aim of dissertation is increasing of convenience of using parking by introduction of automatic car parking system.

In this work we developed the structure of the system, we chose separate elements, we created the model of parking in the packet MATLAB. We developed functional schemes of knots of the system.

Key words: automatic parking, internet of things, IOT, MATLAB, self-conducting cars.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП.....	11
1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	14
1.1 Огляд і класифікація систем паркінгу.....	14
1.2 Проблеми автомобільних парковок	15
1.3 Аналіз існуючих систем керування паркінгом	16
1.3.1 Система керування паркінгом «Dahua».....	16
1.3.2 Система керування паркінгом AFAPARK.....	18
1.3.3 Автоматизована парковка FidPark.....	21
1.4 Висновки до розділу	27
2 СИСТЕМИ ПОЗИЦІОНУВАННЯ В ПРИМІЩЕННІ.....	28
2.1 Огляд систем позиціонування.....	28
2.1.1 Технологія UWB	29
2.1.2 Wi-Fi	30
2.1.3 ZigBee	30
2.1.4 NanoLOC	32
2.1.5 Bluetooth.....	32
2.2 Метод ToF (time of flight)	33
2.3 Метод, заснований на вимірюванні сили сигналу	34
2.4 Висновки до розділу	35
3 ТЕХНОЛОГІЯ ІНТЕРНЕТУ РЕЧЕЙ.....	38
3.1 Концепція технології інтернету речей	38
3.2 Технології для побудови систем інтернету речей	39
3.3 Транспортний засіб до всього (V2X)	39

3.4 Проблеми безпеки	40
3.5 Висновки до розділу	41
4 РОЗРОБЛЕННЯ СИСТЕМИ АВТОМАТИЧНОГО ПАРКУВАННЯ	42
4.1 Розроблення та опис структурної схеми.....	42
4.2 Розроблення та опис функціональної схеми	45
5 МОДЕЛЮВАННЯ СИСТЕМИ АВТОМАТИЧНОГО ПАРКУВАННЯ.....	48
5.1 Огляд системи автоматичного паркування	48
5.2 Опис моделі MATLAB Simulink.....	49
5.2.1 Підсистема контролера автомобіля.....	51
5.2.2 Блок поздовжнього контролера.....	52
5.2.3 Блок бічного контролера	55
5.2.4 Модель авто	58
5.2.5 Результати моделювання.....	62
5.3 Модель навколишнього середовища.....	64
5.4 Поведінковий рівень	66
5.5 Планування руху	70
5.6 Згладжування контуру та генерація траєкторії.....	72
5.7 Виконання повного плану	74
5.8 Маневр паркування	77
5.9 Висновки до розділу	86
6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	87
6.1 Опис ідеї проекту	87
6.2 Технологічний аудит ідеї проекту.....	89
6.3 Аналіз ринкових можливостей запуску стартап-проекту.....	90
6.4 Розроблення ринкової стратегії проекту	95

6.5 Розроблення маркетингової програми стартап-проекту	98
ВИСНОВКИ	101
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	102
ДОДАТОК А	Ошибка! Закладка не определена.
ДОДАТОК Б	Ошибка! Закладка не определена.
ДОДАТОК В	Ошибка! Закладка не определена.
ДОДАТОК Г	Ошибка! Закладка не определена.
ДОДАТОК Д	Ошибка! Закладка не определена.
ДОДАТОК Е	Ошибка! Закладка не определена.
ДОДАТОК Ж	Ошибка! Закладка не определена.
ДОДАТОК З	Ошибка! Закладка не определена.
ДОДАТОК К	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ББМ – Блок маяків Bluetooth low energy

БГП – Блок гальмівних приводів

БЕ – Блок електродвигунів

БК – Блок керування

БКА – Бортовий комп'ютер автомобіля

БКР – Блок керування рухом автомобіля

БС – Базова станція

МП – Мобільний пристрій

ПІ – Пропорційно-інтегральний

ТЗ – Транспортний засіб

ANPR – Automatic Number Plate Recognition

AoA – Angle of arrival

BLE – Bluetooth low energy – Bluetooth з низьким енергоспоживанням

DSRC – Dedicated short-range communications – Виділені комунікації короткого діапазону

GFSK – Gaussian Frequency-Shift Keying – Гаусівська частотна маніпуляція

GLONASS – Global Navigation Satellite System – Глобальна Навігаційна Супутникова Система

GPS – Global Positioning System – Система глобального позиціонування

IMU – inertial measurement unit – Інерційний вимірювальний пристрій

NFER – Near-field electromagnetic ranging – Технологія розповсюдження в ближньому електромагнітному полі

RFID – Radio frequency identification – Радіочастотна ідентифікація

RRT* – rapidly exploring random tree – Швидко вивчає випадкове дерево

RSSI – Received signal strength indicator – Показник рівня сигналу

RTLS – Real-time Locating Systems – Система позиціонування в режимі реального часу

SLAM – simultaneous localization and mapping – Одночасна локалізація і картографування

SPI – Serial Peripheral Interface – Послідовний периферійний інтерфейс

TDoA – Time difference of arrival

ToA – Time of arrival

ToF – Time of flight

TWR – Two-way ranging – Метод двонаправленого вимірювання відстані

UWB – Ultra Wideband – надширокосмугова технологія

V2X – Vehicle-to-everything – Транспортний засіб до всього

ВСТУП

Актуальність теми. Одним з важливих питань для великих міст є питання парковки – найбільш невідпорядковане питання у транспортній системі великих міст. При щорічному збільшенні кількості автомобілів в Україні, зростання кількості паркінгів значно менше. Крім того спостерігається незначна кількість автоматизованих автостоянок.

Неавтоматизовані автостоянки незручні як для клієнтів, так і для адміністрації. Водії змушені петляти темними підземними або багатоярусними автостоянками або об'їжджати цілі квартали в пошуках вільного місця, що викликає лише роздратування та хвилювання за автівку, через надмірно вузькі проїжджі частини іноді трапляються невеликі дорожньо-транспортна пригода (ДТП), що може привести до втрати клієнтів та зайвих проблем. Адміністрація, щоб уникнути заторів або для надання автостоянці хоч якого контролю, зазвичай змушена виділяти регулювальників на в'їзді, що призводить до додаткових витрат і збільшення персоналу.

В Європі автоматизовані паркінги досить давно та міцно зайняли свою нішу. Жоден європейський паркінг не обходиться без систем автоматизації в більшій або меншій мірі.

Очевидно, що без системи інформування про вільні місця на парковці, водій може їздити колами та, не знайшовши вільного місця, може навіть залишити її. Зіткнувшись з подібною ситуацією, в подальшому користувач буде намагатися залишати свій транспортний засіб десь за межами паркінгу аби знову не втрачати час в пошуках місця парковки. Паркування без системи управління можна порівняти з міським центром, транспортні розв'язки якого не обладнані світлофорами. Завдяки цифровими табло, водії орієнтуються про наявність і напрямок до вільних місць прямо при в'їзді на паркінг. Паркінги без контрольованого заїзду на них можуть моніторитись за допомогою камер відео спостереження, що при спеціальному програмному забезпеченні дозволяє вивести кількість вільних місць на карти. Для розкриття потенціалу інфраструктури паркінгів України потребують вдосконалення.

Особливо це стосується технологічного та інформаційного забезпечення та систем контролю, що зменшить клопоту як адміністрації автостоянок так і користувачам.

Використання систем позиціонування – одне з актуальних напрямків вдосконалення технологічних і бізнес процесів в галузях безпілотних автомобілів та систем автоматичного паркування автомобілів.

В більшості випадків ми щодня стикаємося з глобальними навігаційними системами: GPS, GLONASS. Їх робота забезпечується спеціальними супутниками, виведеними на геостаціонарну орбіту, що прив'язані до деякої точки в глобальній системі координат. Пристрої, наприклад, навігатори або мобільні телефони з підтримкою цих систем отримують сигнал супутника і визначають своє місце розташування в прив'язці до географічних координат на основі завантажених у пристрій карт місцевості. Точність роботи глобальних навігаційних систем достатня в масштабах великих територій, мова йде про кілометри або ж сотні метрів в масштабах кварталів, міста або країни, та іноді похибка може досягати 5-20 метрів в залежності від перешкод та якості сигналу.

У випадку з підземними паркінгами або паркінгами всередині великих торговельних центрів, точно визначити місце знаходження за допомогою глобальних систем практично неможливо. Залізобетонні будівлі, підвали, тунелі, сильно впливають на кінцеву точність сигналу, також можуть впливати несприятливі погодні умови та інші подібні фактори.

Саме цих недоліків позбавлені системи локального позиціонування, їх вкрай очевидні переваги в порівнянні з глобальними навігаційними системами стають у нагоді при побудові систем локації та зв'язку в межах певних територій і приміщень, а саме в межах підземних паркінгів, що нас і цікавить.

Системи локального позиціонування здатні вирішувати завдання з визначенням місцезнаходження людей, техніки, інших фізичних об'єктів в невеликих масштабах. В межах цілком конкретної території або всередині будівлі з високою точністю приблизно (від 5м до 0.1м). Такі комплекси зазвичай складаються із стаціонарних та пристроїв які знаходяться на рухомих об'єктах, блоку(ів)

керування, сервера і спеціального програмного забезпечення, структура може видозмінюватись в залежності від потреб.

Тому тема дипломного проекту, яка присвячена створенню розумної системи паркування є актуальною.

Метою дослідження є підвищення ефективності використання паркінгів за рахунок впровадження «розумної» системи паркування.

Задачі дослідження. Для досягнення поставленої мети визначені наступні задачі, які необхідно вирішити:

- 1) дослідити існуючі рішення в даній області;
- 2) розробити структурну схему системи;
- 3) розробити функціональну схему системи;
- 4) розробити схему взаємодії авто з системою;
- 5) побудувати модель «розумного» паркінгу;
- 6) дослідити роботу системи на моделі.

Об'єктом дослідження є паркінг.

Предмет дослідження. «розумна» система паркування.

1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд і класифікація систем паркінгу

Автомобільна парковка – місце стоянки транспортних засобів, що представляє собою ділянку проїзної частини автомобільної дороги, вулиці населеного пункту або прилеглої до них території.

Парковка – невід'ємна частина житлового, офісного, адміністративного комплексів, а також торгово-розважальних центрів. Сьогодні практично для кожного об'єкта будівництва, так чи інакше, вирішується питання розміщення автомобілів його відвідувачів.

Автоматизована паркувальна система (АПС) – складається зі спеціалізованого обладнання та програмного забезпечення. АПС дозволяє проїзд автотранспорту на парковку і здійснює збір оплати за послуги паркування в автоматичному режимі, тобто, мінімально вдаючись до допомоги людини. Спеціалізованим обладнанням вважаються стійки в'їзду і виїзду, шлагбауми (пропускають автотранспорт на територію), автоматична каса або ручна каса (для оплати), система визначення номерів (відеокамери, плати відео захвату).

Інтелектуальна система паркування (IoT Parking) – складається з спеціалізованого обладнання та програмного забезпечення. Дозволяє проїзд автотранспорту на парковку, автоматично паркує авто, тобто, без участі водія, автоматично розраховує оплату за перебування авто на парковці по кінцевому часу перебування на паркінгу. Спеціалізованим обладнанням вважаються: стійки в'їзду і виїзду, шлагбауми що пропускають автотранспорт на територію; автоматична каса; інтернет банкінги, термінали, електронні гаманці т.і.; модулі безпроводного зв'язку: BLE маяки для локального позиціонування або інші подібні системи, Wi-Fi модулі, Bluetooth модулі, 4G/5G модулі; датчики: датчики руху, датчики освітлення, датчики зайнятості місць; контролер управління рухом автомобіля; сервери: даних, додатків, сервісів.

Специфіка таких паркінгів полягає в постійному переміщенні авто транспорту без участі водія, що і диктує основні вимоги до транспортних засобів:

- наявність безпілотного керування в авто;
- наявність паркувальних систем (Park Assist);
- наявність різних давачів по типу (лідари, парктроніки, т.і.);
- можливість прийому-передачі даних від авто до системи.

1.2 Проблеми автомобільних парковок

Київ, за кількістю автомобілів наздогнав найбільші міста Європи, задихається без необхідної кількості паркувальних місць. Рятуючи місто від транспортного колапсу, влада створює перехоплюючі паркінги. Один з таких знаходиться біля метро «Видубичі».

Перехоплюючий паркінг – автостоянка, сполучена з громадським транспортом, що дозволяє жителям приміської зони та іншим людям вирушати в центр міста, полишивши свої транспортні засоби та пересівши в автобус, маршрутне таксі, трамвай, метро тощо на решту подорожі. Автомобіль зберігається на стоянці протягом дня і забирається власником на зворотньому шляху. Перехоплюючі стоянки, як правило, розташовані в під'їзних зонах до центру великих міст або в передмісті [1].

Вже не перший рік місто задихається в пробках, а пошуки парковки іноді перетворюються для водія в нездійсненне завдання. Вся справа в тому, що Київ, з самого початку не розрахований на велику кількість машин. За даними Zik.ua, у 2015 на 1 тис. Киян припадало 213 автівок, а у 2017 – 257. Зараз ця цифра доходить до 300, а за підрахунками «А+С Україна» до 2025 року сягне 375 авто на 1000 жителів. Київ уже наздогнав Нью-Йорк за кількістю авто [2].

Парковка автомобілів на сьогоднішній день є найбільш складним завданням для автовласника. Найчастіше дуже складно відразу зорієнтуватися у виборі вільного паркувального місця, при великому скупченні автомобілів. Крім того, ситуація на парковці постійно змінюється, і зайняті місця можуть в будь-який момент звільнитися. Втім складність не тільки у знаходженні вільного місця, майже

кожен водій, а особливо мало досвідченні, стикаються з проблемою паркування з браком вільного місця для маневру. Іноді складно відразу зрозуміти чи достатньо вільного місця чи ні.

1.3 Аналіз існуючих систем керування паркінгом

1.3.1 Система керування паркінгом «Dahua»

Система управління паркуванням Dahua [3] – це навігаційна система, яка розроблена для завантажених і багаторівневих паркінгів, де водії часто змушені їздити колами в пошуках вільного місця. Система інформування про вільні місця на парковці забезпечує супровід водіїв, в реальному часі направляючи їх до вільних місць за рівнями, зонам і смугам. Система працює з будь-яким типом чотириколісних транспортних засобів.

Які проблеми вирішує дана система:

- труднощі пошуку вільних місць на парковці;
- ілюзорна завантаженість паркінгу при наявності вільних місць;
- аварії і затори, як наслідок хаотичності парковки.

Система інформування про вільні місця на парковці працює наступним чином.

Датчик, розроблений спеціально для визначення наявності під ним транспортного засобу, а також світлодіодний індикатор розміщені безпосередньо над кожним паркомісцем. Щомиті датчик зчитує наявність транспортного засобу і передає свій статус в центр управління через структуровану кабельну систему (СКС). ПК обробляє дані і оновлює інформацію на екранах навігації (цифрових табло), надаючи водіям дані про вільні місця в реальному часі.

Як тільки водій стикається з необхідністю вибору напрямку руху (рівень, зона або ряд), цифрові табло, за допомогою динамічних стрілок, показують шляхи для руху. Після цього необхідно просто дотримуватися вказівниками для визначення вільного паркувального місця в найкращих умовах і в найкоротші терміни.

Систему управління паркуванням Dahua, яка зображена на рисунку 1.1, відрізняють:

- повністю автоматична система управління;
- розумна вартість системи;
- безвідмовна, надійна робота;
- мінімум часу для пошуку місця на парковці;
- спостереження за допомогою камер високої якості.



Рисунок 1.1 – Система управління парковкою Dahua [3]

На рисунку 1.2 зображено систему ідентифікації водія або авто.

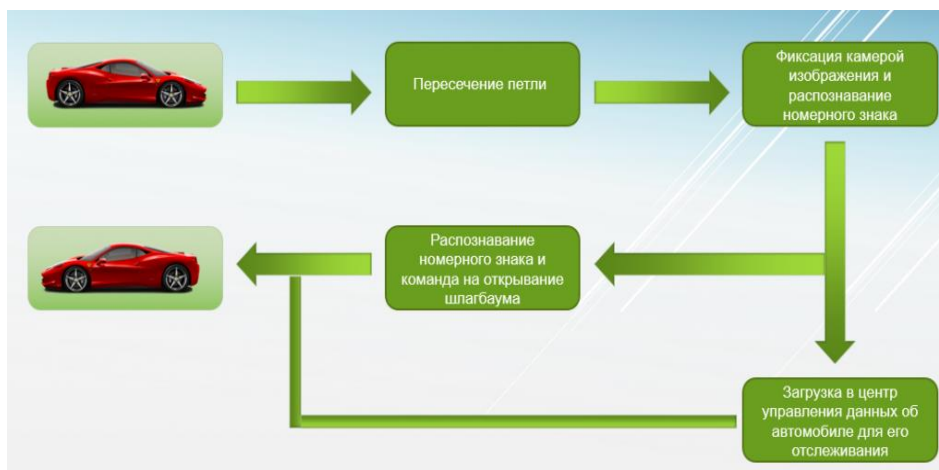


Рисунок 1.2 – Система ідентифікації водія/авто [3]

До появи системи інформування про вільні місця на парковці, проектувальники повинні були залишати вільний проїзд в кінці кожного з рядів, щоб запобігти утворенню пробок в години найбільшого завантаженості паркінгу. З появою системи управління необхідність в таких вільних проїздах відпала. Це дозволяє зробити паркінг більш прибутковим, за рахунок збільшення кількості

паркомісць.

На рисунку 1.3 наведено зразок, того як система управління паркуванням дозволила збільшити місткість паркінгу на кожному рівні, що істотно збільшує кількість паркомісць при однакових затратах на будівництво будівлі.

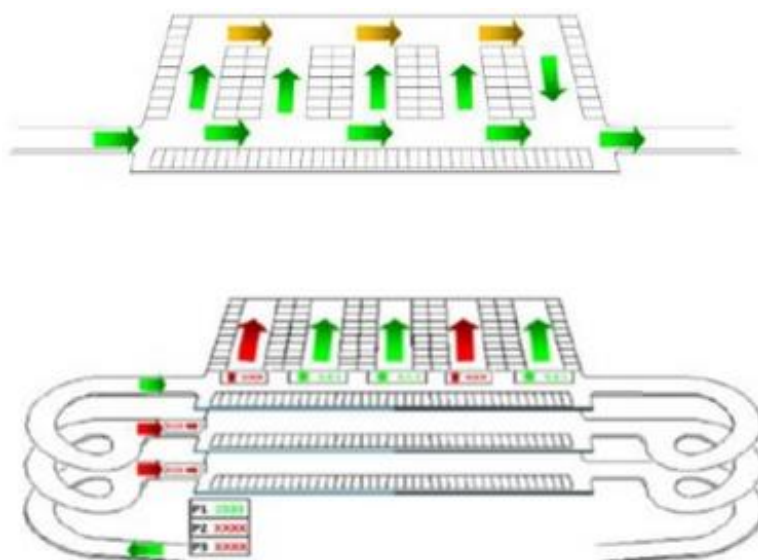


Рисунок 1.3 – Зразок розташування паркомісць [3]

Часто буває, що транспортний засіб залишається на стоянці автомобілів довше дозволеного терміну. Знання того, де вони припарковані, означає, що можна визначити їх реєстраційний номер. В такому випадку, клієнту, який стверджує, що втратив свій паркувальний квиток і каже, що він заїхав годину назад, може бути викрито.

1.3.2 Система керування паркінгом AFAPARK

Визначення та інтуїтивна індикація вільних паркувальних місць у реальному часі, динамічні і інтуїтивні інформаційні табло з відео і аудіо інформацією дозволяють автомобілістам вільно рухатися по парковці. Рух користувачів проходить в комфортній обстановці починаючи з в'їзду на парковку до місця стоянки.

При в'їзді на паркінг динамічний дисплей показує загальну кількість вільних місць для паркування і запрошує водія в'їжджати, інформуючи його про наявність місць.

В середині парковки динамічні покажчики напрямку спрощують рух автомобілісту, показуючи на якому рівні і в якій зоні знаходяться вільні паркувальні місця.

У кожній зоні динамічні покажчики показують де знаходяться вільні місця.

Після вибору водієм автомобіля алеї, миготлива сигналізація зеленого кольору чітко показує знаходження вільного місця. У момент парковки авто, датчик, найчастіше розташований зверху на стелі або в центрі парко-місця, визначає присутність автомобіля і відразу ж змінює колір із зеленого на червоний.

Дисплеї спочатку алеї і в даній зоні відразу ж віднімають дане місце з наявності вільних місць.

Як додаткова послуга, в корпус датчика може бути вбудований звуковий гучномовець, який після визначення наявності автомобіля, автоматично включає звукове персоналізоване повідомлення. Це повідомлення допомагає власнику автомобіля швидко запам'ятати номер парко-місця. Після цього звукового повідомлення може бути включено додаткове інформаційне або рекламне повідомлення, персоналізоване дистанційно вашим відділом маркетингу за допомогою спеціальної вбудованої програми. Таке ж звукове повідомлення може бути включено для спеціалізованих місць для паркування (інвалідних, сімейних, для електромобілів і т.д.)

Як додаткова послуга в сигналізацію може бути вбудований звуковий гучномовець, який автоматично включає приємну музику, яка надає гармонійну обстановку паркінгу. Гучномовець може в цей же час поширювати повідомлення про заходи евакуації при пожежі.

AFAPARK постійно розробляє і вдосконалює свій асортимент продукції, адаптуючись до різних архітектурних обмежень парковок, і тим самим дозволяє оптимізувати витрати на впровадження системи.

Робота системи з точки зору клієнта відбувається наступним чином [4].

При в'їзді на паркінг, завдяки світловим табло, встановленим вздовж проїзду, можливо дізнатися про наявність вільних місць або ж їх відсутності. При в'їзді в сектор яскрава світлова сигналізація показує усі вільні паркувальні місця,

забезпечуючи швидку і комфортну парковку. Сигналізатор добре помітний незалежно від напрямку руху. Після зупинки автомобіля система озвучування паркувального місця повідомляє пасажирів про їх місцезнаходження, щоб допомогти знайти машину, або ж для попередження про те, що дане місце призначене для інвалідів.

Керуючий персонал паркінгу бачить роботу системи наступним чином.

Крім комфорту для клієнта візуально-звукова система адміністрування та управління рухом на автостоянці AFAPARK забезпечує адміністратору ефективне управління паркінгом і скорочення витрат завдяки графічній і статичній інформації. В аварійній ситуації адміністрація зможе повідомити про евакуацію. Також система AFAPARK може використовуватися як джерело додаткового доходу, наприклад, для демонстрації реклами. Використання даної системи підвищує рентабельність.

Вигода з економічної точки зору від використання системи організації, керівництва та управління паркінгом AFAPARK здобувається декількома способами.

Скорочення кількості переміщень по паркінгу оптимізує рух. З системою AFAPARK все вільні паркувальні місця будуть позначені точно і зрозуміло. Також відбувається скорочення часу, що витрачається на парковку, в середньому на 6 хвилин.

На більшості паркінгів навіть при найбільш оптимальному заповненні не задіяне і близько 2% паркувальної площі. Застосування системи AFAPARK гарантує використання 100% площі, так як, завдяки зрозумілій візуалізації, водії отримують найсвіжішу інформацію про поточну ситуацію на паркінгу, а система направляє їх на найближче вільне місце.

Екологічна цінність даної системи спостерігається в декількох моментах, описаних нижче.

За допомогою системи AFAPARK машини прямують безпосередньо на вільні місця, а час переїзду скорочується до мінімуму. Істотно знижується рівень витрати палива і забруднення повітря, через що клієнти і адміністратори паркінгу знаходяться в більш екологічно чистих умовах навколишнього середовища. Скорочення забруднень зменшується до 20%, а це дуже значуща цифра.

Економічна вигода від використання системи організації та управління паркінгом AFAPARK відчутна вже на етапі проектування.

Система AFAPARK це :

- гарантована 100% зайнятість стоянкових місць;
- зниження (до 10%) інвестиційних витрат при будівництві автостоянки;
- економія часу (середній час паркування 2 хвилини замість 8 хвилин), завдяки візуальній та звуковій наводкою на парковочне місце;
- полегшений рух по території автостоянки;
- зменшений ризик втрати автомобіля і підвищена безпека;
- ідентифікація місць стоянок, відведених для інвалідів, або іншого призначення (хвилинна стоянка, місця для VIP тощо);
- зниження витрат експлуатації паркінгу (економія електроенергії: вентиляція і освітлення);
- зниження (майже 4-кратне) емісії вихлопних газів (CO₂), ЗНАЧНЕ зменшення забруднення з користю для працівників і користувачів.

1.3.3 Автоматизована парковка FidPark

FidPark-CAR призначена для автоматизації платних і службових автостоянок. Система надає водіям різні способи оплати за стоянку – передоплата, депозити, квитки заходів, різні абонементи. Програмне забезпечення FidPark дозволяє "налаштувати" параметри автостоянки для досягнення заданої бізнес-цілі [5]:

- отримання максимального прибутку автостоянки;
- максимальна пропускна здатність об'єкта, який вона обслуговує;
- постійна наявність вільних місць.

Характеристики та можливості FidPark:

- необмежена кількість контрольних пунктів (в'їздів і виїздів);
- необмежена кількість паркувальних зон з різними тарифами та контролем доступу;
- можливість розширення функціональності системи та її інтеграції з

системами обліку;

- необмежена кількість користувачів;
- необмежена тривалість паркування автомобілів;
- необмежена кількість постійних клієнтів;
- необмежена кількість тарифних планів;
- різні тарифи на кожен день тижня;
- установка цін для кожного інтервалу тривалістю 5 хв. - 24 години;
- перелік святкових днів і робочих днів-винятків;
- збереження локальної функціональності при розриві зв'язку з сервером;
- захист від вандалізму і злому програмного забезпечення;
- відповідність фіскального законодавства;
- зручна інтеграція з зовнішніми WEB платіжними системами;
- система знижок, яка працює через WEB;
- можливість інтеграції з системою контролю доступу.

У версії FidPark-ANPR з автоматичним розпізнаванням номерів (Automatic Number Plate Recognition) автомобільний номер розпізнається спеціальними відеокамерами і використовується або як доповнення до інших засобів ідентифікації (RFID, штрих-код і т.п.), або замість них. ANPR значно розширює можливості FidPark і FidPark +, а також підвищує зручність використання системи водіями. Наприклад, з'являється можливість автоматичної оплати кожної стоянки з банківської картки.

Функцію ANPR можна використовувати як додаткову опцію. Приклад роботи цієї функції зображений на рисунку 1.4.



Рисунок 1.4 – Приклад роботи функції ANPR [5]

Якщо ANPR використовується як додаткова опція, FidPark набуває таких можливостей:

- якщо клієнт з одноразовим штрих-кодовим талоном заплатив за паркування, то система FidPark може потім випустити цю машину без необхідності сканування цього талона на виїзді, за винятком рідкісних випадків помилкового розпізнавання номера;
- адміністратор може легко визначити дату / час заїзду машини з конкретним номером і відповідну вартість парковки, що вирішує проблеми розрахунків з клієнтами, що заявили про втрату паркувального талона довготривалої стоянки;
- порівняння номерів на в'їзді і виїзді запобігає виїзду автомашини за чужою перепусткою;
- на багатозонних парковках ANPR – камери між зонами дають можливість без додаткових терміналів і бар'єрів контролювати фактичне перебування автомобіля в кожній зоні;
- платежі зареєстрованих клієнтів, прив'язані до номера їхніх автомобілів, автоматично списуються з їх картки або мобільного рахунку, що звільняє їх від необхідності платити касиру або в касовому автоматі;
- клієнт може забронювати місце на стоянці, ввівши на смартфоні реєстраційний номер свого автомобіля замість отримання і використання талона зі штрих-кодом;
- з'являється можливість побудувати службову автостоянку без в'їзних і виїзних терміналів, що має велике значення в містах з жорсткими правилами будівництва та землекористування.

Функцію ANPR використовують для постійних клієнтів. Приклад роботи функції представлений на рисунку 1.5.



Рисунок 1.5 – Приклад роботи функції ANPR для постійних клієнтів [5]

FidPark з ANPR як єдиною технологією ідентифікації, найбільш зручна для службових автостоянок, і в міру зниження цін на відеокамери стає все більш популярною.

Нижче наведено перелік можливостей FidPark-ANPR:

- всі клієнти звільняються від необхідності будь-яких дій з використанням талонів, карток і т.п.;
- усунення паперових талонів зі штрих кодами для разових клієнтів дозволяє знизити витрати на підтримку та технічне обслуговування системи;
- усувається можливість зловживань з багаторазовим використанням одного пропуску (antipassback);
- зареєстровані клієнти з відомими реєстраційними номерами автомашини можуть використовувати стоянку, не витрачаючи час на оплату, яка при виїзді буде зніматися автоматично з рахунку клієнта;
- клієнт може забронювати місце для паркування через Інтернет, просто ввівши реєстраційний номер свого автомобіля;
- у разі паркування без шлагбаумів (on-street) забезпечується автоматична реєстрація часу в'їзду / виїзду і генерації відповідних звітів / рахунків для

справляння оплати з клієнтів.

Приклад використання функції ANPR разовими клієнтами наведений на рисунку 1.6.

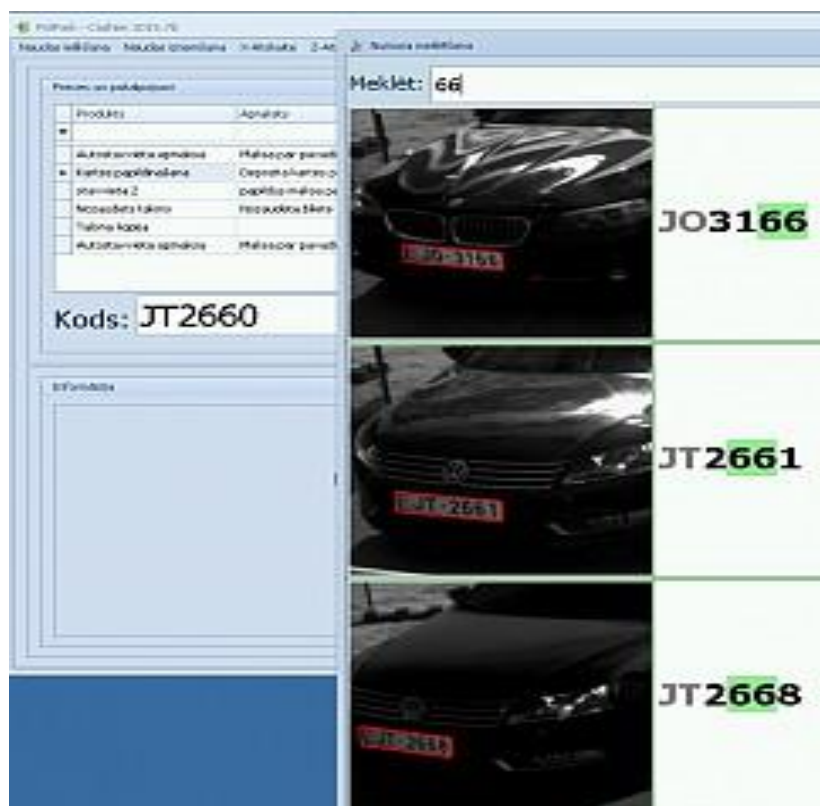


Рисунок 1.6 – Приклад роботи функції ANPR для разових користувачів [5]

Особливу увагу в версії системи FidPark-ANPR для платних стоянок було приділено тому, щоб зробити все операції більш доброзичливими для клієнта навіть у разі, якщо один або декілька символів номерного знаку були розпізнані помилково. В іншому випадку процедура пошуку номера може зайняти багато часу і створити стрес у клієнта – особливо для людей, які не мають досвіду роботи з сенсорними екранами. В автоматичній касі клієнту достатньо ввести декілька цифр номерного знаку, щоб вибрати зі списку свою автомашину для оплати стоянки. Аналогічна процедура реалізована і в програмі касира.

Система може ідентифікувати клієнтів за допомогою багатьох різних способів. FidPark підтримує такі технології ідентифікації клієнтів як:

- безконтактні мітки (картки, браслети і т.п.) з відстанню зчитування до 10 см (125 KHz, 13,56 Mhz, ISO / IEC 14443a / b);

- безконтактні мітки (картки, брелоки) з відстанню зчитування до 1 м (ISO / IEC 15693);
- безконтактні мітки з відстанню зчитування до 10 метрів (UHF);
- NFC-картки або мобільні телефони з відстанню зчитування до 2-3 см;
- 2D штрих-коди на разових талонах або екрані смартфона;
- держномер автомобіля;
- номер або IMEI мобільного телефону (смартфону);
- відбитки пальця, інші методи біологічної ідентифікації.

Конкретний вибір пристроїв і технологій визначається специфікою об'єктів і побажаннями замовників. Допустимо використовувати кілька ідентифікаторів одного клієнта (наприклад, картку і браслет).

Інформація про клієнтів і події зберігається тільки в базі даних FidPark, що забезпечує високу захищеність даних про клієнтів при помірних витратах, оскільки зчитується тільки унікальний ідентифікатор. Висока відмовостійкість реалізована за рахунок використання процесорів в кожному пристрої (терміналі).

Використання мобільних додатків для ідентифікації водія.

В системі FidPark-Phone для ідентифікації водія використовується мобільний телефон. На в'їзді водій повинен зателефонувати з нього на певний номер (зазвичай зображений на стрілі шлагбаума), дзвінок є безкоштовним. Якщо власник номера телефону має право на в'їзд, то шлагбаум відкривається, в іншому випадку на телефон через кілька секунд надходить SMS із зазначенням причин відмови.

FidPark-Phone найбільш зручно використовувати на службових автостоянках, куди дозволений в'їзд тільки постійним клієнтам або відвідувачам за попереднім погодженням. При цьому зберігається функціональність системи FidPark, зокрема – групові пропуски, тайм-зони, захист від багаторазового в'їзду.

Для управління системою можна використовувати будь-який Інтернет-браузер. Користувачі (орендарі) автостоянки в межах виділених їм квот (кількість місць, тайм-зони) можуть самостійно редагувати список номерів телефонів водіїв, які мають право в'їзду на автостоянку, а також реєструвати "своїх" відвідувачів, вводячи номери їх мобільних телефонів і час початку візиту (в'їзду на стоянку). При цьому

система може обчислити вартість стоянки відвідувача по її тривалості, і віднести дану суму на рахунок користувача стоянки.

FidPark-Phone можна ефективно використовувати і на платних автостоянках як альтернативу системі FidPark-LPR з розпізнаванням номерів.

1.4 Висновки до розділу

Проаналізувавши низку прикладів, можна отримати певні результати.

Було розглянуто задачі, для яких використовують автоматизацію паркінгу, а саме:

- Труднощі пошуку вільних місць на парковці;
- Ілюзорна завантаженість паркінгу при наявності вільних місць;
- Аварії і затори, як наслідок хаотичності при зайнятості парко-місць;
- Мінімізація в часі для пошуку парко-місць.

Також була розглянута реалізація візуалізації вільних місць у реальному часі.

Динамічні та інтуїтивні інформаційні табло з відео та аудіо інформацією, що дозволяють здійснювати без проблемний та безперешкодний рух по паркінгу.

Отже, на швидкодію системи впливає велика кількість факторів:

- максимальна пропускна здатність об'єктів, які вона обслуговує;
- постійна наявність вільних місць;
- велика кількість зон для паркування;
- ширина проїзної частини;
- різноманітність парко-місць;
- збереження локальної функціональності при розриві зв'язку;
- зручна інтеграція з зовнішнім WEB платіжними системами.

Можливість розширення функціональності системи та її інтеграції з системами обліку.

Все це показує, на скільки різноманітними можуть бути паркінги.

2 СИСТЕМИ ПОЗИЦІОНУВАННЯ В ПРИМІЩЕННІ

2.1 Огляд систем позиціонування

Глобальна система позиціонування (GPS) зазвичай використовується для навігації в зовнішніх середовищах. Однак вона недоступна для позиціонування в приміщеннях через непрохідність сигналів. В результаті чого широко вивчаються різні підходи до навколишнього середовища, що забезпечують надійність та високу точність позиціонування для людини або пристрою в приміщенні. Дослідники пропонують багато варіантів використання радіомаяків, інерційних датчиків, ультразвуку, зору тощо. Більшість систем внутрішнього позиціонування поділяються на два типи: не інфраструктурні та інфраструктурні.

Використання систем локального позиціонування для різних об'єктів – одна з актуальних тем у різних галузях діяльності суспільства. У різних ситуаціях потрібні знання як визначати місцезнаходження в певний момент часу (в режимі реального часу). Для цього проміжок часу між вимірами повинен бути таким, щоб об'єкт, під час руху з характерною для нього швидкістю, встигав проходити відстань не більше подвоєної точності позиціонування. Для прикладу, щоб забезпечити позиціонування (в реальному часі) з точністю в 1 м. для людини, що має швидкість переміщення 5,4 км/год або ж (1,5 м/с), виміри потрібно проводити з періодичністю приблизно не менше одного разу кожні 1,3 секунди [6].

Використання RTLS (систем визначення місця розташування в режимі реального часу) для систем розумних паркінгів залежить від поставлених завдань і цілей.

Найбільшою групою, що поділяється ще на кілька підгруп, є радіолокаційні технології. Ті ж, в свою чергу, діляться на стандартні технології передачі даних (Wi-Fi, Bluetooth, ZigBee), так чи інакше призначені для вимірювання відстаней, та на ті, які, виходячи з фізичних властивостей модуляції, найбільш підходять для вимірювання відстаней (UWB, NFER та ін.) [6].

2.1.1 Технологія UWB

Технологія UWB (Ultra Wideband – надширокосмугова) використовує короткі імпульси з максимальною пропускнуою здатністю при мінімальній центральній частоті. У більшості виробників центральна частота становить кілька гігагерц, а відносна ширина смуги – 25-100%. Технологія використовується у зв'язку, радіолокації, вимірюванні відстаней і позиціонуванні [6].

Виробники пропонують різні варіанти UWB технології. Розрізняються форми імпульсів. У деяких випадках використовуються щодо потужні поодинокі імпульси, в інших – сотні мільйонів малопотужних імпульсів в секунду. Застосовується як когерентна (послідовна) обробка сигналу, так і некогерентна. Все це призводить до значної розбіжності характеристик UWB систем різних виробників [6].

Принципові обмеження UWB технології [6-7]:

- важко побудувати систему з істотною потужністю передачі (типова потужність передавача – 50 мкВт, дуже потужного – 10 мВт);
- обмеження з боку органів частотного регулювання (в зв'язку з цим системи, як правило, використовуються в приміщеннях, де їх малопотужний шумоподібний сигнал не впливає на інші системи і навіть не виявляється).

Переваги:

- високий рівень перешкодозахищеності;
- висока безпека;
- чим вище частота, тим більше точність (але тим менше радіус дії).

Недоліки:

- малий радіус дії (до 10 м);
- складна інфраструктура.

Методи визначення позиції: AoA, ToF, ToA, TDoA.

2.1.2 Wi-Fi

Wi-Fi – це технологія передачі даних середнього радіусу дії, зазвичай покриває десятки метрів, яка використовує діапазони частот, що не ліцензується, для забезпечення доступу до мережі. Оскільки Wi-Fi з самого початку не призначався для використання в якості технології локального позиціонування, стандартна мережа надає інформацію з точністю лише до точки доступу, тому для підвищення точності визначення місцеположення використовується RSSI або при деяких допрацюваннях інші спеціалізовані методи (наприклад, TDoA) [6-7].

Переваги:

- широке розповсюдження;
- низька вартість обладнання.

Недоліки:

- для підвищення точності, потрібне збільшення щільності розташування базових станцій;
- завантаженість ефіру Wi-Fi;
- недостатня точність визначення місця розташування для ряду завдань, навіть при застосуванні спеціальних розширень Wi-Fi (в ідеальних умовах 3-5 метрів, в реальності 10-15 метрів).

Методи визначення позиції: RSSI, на основі TDoA.

2.1.3 ZigBee

ZigBee – стандарт для набору високорівневих протоколів зв'язку, що використовують невеликі, малопотужні цифрові трансивери, заснований на стандарті IEEE 802.15.4 для бездротових персональних мереж [7-8].

Технологія ZigBee дозволяє створювати бездротові мережі з автоматичною ретрансляцією повідомлень, що самоорганізуються та самовідтворюються. ZigBee призначений для мобільних пристроїв, що вимагають гарантованої безпечної

передачі даних при відносно невеликих швидкостях і можливості тривалої роботи мережевих пристроїв від автономних джерел живлення (батарейок) [7-8].

Стандарт ZigBee передбачає використання частотних каналів в діапазонах 868 МГц, 915 МГц і 2,4 ГГц. Найбільші швидкості передачі даних і найвища стійкість досягаються в діапазоні 2,4 ГГц. Тому більшість виробників мікросхем випускають приймачі саме для цього діапазону, в якому передбачено 16 частотних каналів з кроком 5 МГц [9].

Швидкість передачі даних разом зі службовою інформацією в ефірі становить 250 кбіт/с. При цьому середня пропускна спроможність вузла для корисних даних, в залежності від завантаженості мережі і кількості ретрансляцій, може лежати в межах 5 ... 40 кбіт/с [9].

Відстані між вузлами мережі становлять десятки метрів під час роботи всередині приміщень і сотні метрів на відкритому повітрі. За рахунок ретрансляції зона покриття мережі може значно збільшуватися [9].

Переваги [7]:

- підтримує як прості топології мережі («точка-точка», «дерево» і «зірка»), так і mesh-топологію з ретрансляцією і маршрутизацією повідомлень;
- містить можливість вибору алгоритму маршрутизації, в залежності від вимог програми та стану мережі;
- простота розгортання, обслуговування та модернізації;
- здатність до самоорганізації та самовідтворення;
- низький рівень споживання енергії.

Недоліки:

- низька швидкість передачі даних.

Методи визначення позиції: RSSI, TDoA, ToF.

2.1.4 NanoLOC

NanoLOC – це технологія компанії Nanotron Technologies, багато в чому схожа з більш старою версією NanoNET. Крім швидкості передачі інформації в 1 Мбіт в секунду на відстані в кілька сотень метрів, ця технологія дозволяє визначати відстань між приймачами. Похибка у визначенні відстані – 2 метри, що дозволяє визначати, де знаходиться приймач-передавач по відношенню до інших таких приймачів-передавачів. Якщо необхідно визначення в тривимірній системі координат, знадобляться чотири (і більше) передавача NanoLOC, координати розташування яких вже відомі [7].

Переваги [7]:

- можливість роботи в неліцензованих діапазонах частот при потужності до 100 мВт;
- методи, що використовуються для визначення місця розташування забезпечують можливість локалізації об'єктів за межами периметра зони обслуговування зі зниженням точності;
- великий вибір готового ПО (з відкритими кодами);
- автокореляційні властивості сигналу роблять технологію стійкою до зовнішніх перешкод.

Недоліки:

- обмеження за кількістю пристроїв в сегменті;
- пропрієтарна технологія.

Методи визначення позиції: RSSI, TDoA, ToF, TWR.

2.1.5 Bluetooth

Bluetooth – специфікація бездротових персональних мереж ближнього радіусу дії, яка працює в частотному діапазоні 2,4-2,4835 ГГц. У Bluetooth несуча частота сигналу змінюється 1600 разів в секунду псевдовипадковим чином, це дозволяє

уникнути проблем при функціонуванні групи пристроїв в безпосередній близькості, а також підвищити безпеку передачі даних [7].

Переваги [7].

- підвищена безпека і перешкодозахищеність;
- низький рівень споживання енергії (BLE);
- недороге обладнання;
- компактність модулів.

Недоліки:

- неможливість досягнення високої точності визначення місця розташування.

Методи визначення позиції: RSSI.

2.2 Метод ToF (time of flight)

Метод ToF (time of flight) заснований на вимірюванні часу, який витрачає електромагнітна хвиля на подолання відстані між базовою станцією (БС) до мобільного пристрою (МП) і повернення до БС після його відбиття від МП. Отримавши цей час і знаючи швидкість руху хвилі, можна обчислити відстань від мобільного пристрою до БС:

$$S = \frac{1}{2} \cdot t \cdot c,$$

де t – час подолання хвилею відстані від БС до МП,

c – швидкість розповсюдження радіохвилі (швидкість світла).

Переваги:

- мобільний пристрій – пасивний (не споживає енергію);
- висока точність визначення місця розташування мобільного пристрою;
- великий радіус дії;
- відсутність необхідності синхронізації між пристроями.

Недоліки:

- МП має відображати електромагнітну хвилю в напрямку БС;
- неможливо розділити мобільні пристрої (для ідентифікації) [13].

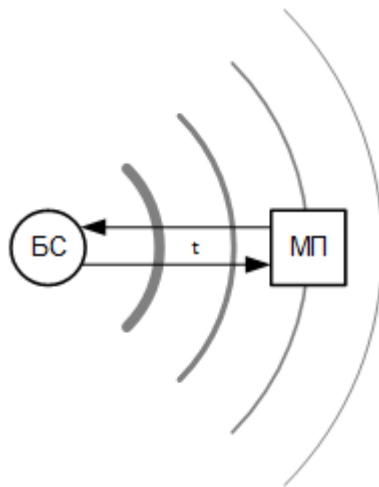


Рисунок 2.1 – Метод ToF [13]

2.3 Метод, заснований на вимірюванні сили сигналу

RSSI (received signal strength indicator) – це індикатор рівня потужності сигналу. Метод дозволяє визначити місце розташування пристрою, ґрунтуючись на рівні інтенсивності сигналу, отриманому БС або навпаки. Для використання цього методу застосовується або перетворення рівня потужності сигналу в відстань, або встановлюються відповідності до карт покриття [13].

Даний метод добре працює на малих відстанях, але при збільшенні дальності дає велику похибку відповідно до специфіки поширення радіосигналу.

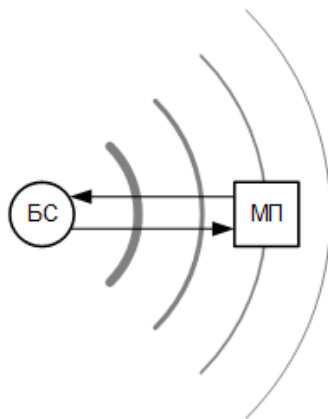


Рисунок 2.2 – Метод заснований на вимірюванні сили сигналу (RSSI) [13]

Переваги:

- мале енергоспоживання мобільним пристроєм;
- низька вартість.

Недоліки:

- низька точність визначення місця розташування мобільного пристрою;
- складні алгоритми визначення місця розташування мобільного пристрою.

2.4 Висновки до розділу

Для позиціонування об'єктів більшість систем використовують комбінацію методів. Даний підхід дозволяє уникнути недоліків одного методу, та доповнити перевагами іншого. Також, виходячи з цілей для вирішення різних задач можуть знадобитися різні методи. Методи вимірювання відстані залежать від наявного обладнання та наявності синхронізації між базовою станцією та об'єктом.

Основні характеристики радіочастотних технологій наведено в таблиці 2.1.

Таблиця 2.1 – Основні характеристики радіочастотних технологій [7]

Технологія	Діапазон робочих частот	Зона охоплення	Точність	Вартість обладнання	Пропускна спроможність
UWB	3-10 ГГц	До 40 м	До 0,1 м	середня	До 480 Мбіт/с
Wi-Fi (IEEE 802.11a/g/n)	2,4 ГГц, 5 ГГц	До 50 м	До 3-5 м	низька	802.11ac: до 6.77 Гбіт/с
ZigBee (IEEE 802.15.4)	868 МГц, 915 МГц, 2,4 ГГц, 6 ГГц	До 300 м	До 2 м	низька	До 250 Кбіт/с

Технологія	Діапазон робочих частот	Зона охоплення	Точність	Вартість обладнання	Пропускна спроможність
NanoLOC (IEEE 802.15.4a)	2,4 ГГц	До 900 м	До 1 м	середня	До 2 Мбіт/с
Bluetooth (IEEE 802.15.1)	2,4 ГГц	До 150 м	До 5 м	низька	До 24 Мбіт/с

Оскільки всі технології вище перераховані окрім Wi-Fi та Bluetooth буде важко розгорнути в умовах паркінгу через його специфіку (постійний кругообіг авто), тому саме ці технології будуть розглянуті для вирішення завдань локального позиціонування на паркінгах. По скільки ефір Wi-Fi буде майже постійно завантажений передачею чималих об'ємів медіа інформації (відео, данні з інших давачів), то доцільніше всього буде використовувати мережу Bluetooth-mesh [14].

Бездротова локалізація в приміщенні за допомогою Bluetooth Low Energy (BLE) маяків привернула значну увагу після виходу протоколу BLE. У визначенні протоколу BLE на передачу повідомлень використовується 40 каналів, кожен з яких становить 2 МГц, навколо радіочастот 2,4 ГГц. Тривалість передачі повідомлень дуже коротка для економії заряду акумулятора. Сила прийнятого сигналу (RSS) з цих трьох каналів може бути використана для оцінки місця розташування цілі. Потужність передачі для BLE-маяків також встановлюється від 0 дБм до -40 дБм. Щоб зменшити споживання енергії. У порівнянні з локалізацією Wi-Fi, локалізація BLE має такі переваги: BLE сигнали RSS можуть мати більшу частоту вибірки, ніж сигнали WiFi RSS (0,25 Гц ~ 2 Гц); BLE споживає менше енергії, ніж Wi-Fi; Сигнали BLE RSS можуть бути отримані з більшості розумних пристроїв, тоді як сигнали WiFi RSS не можуть бути надані портативними пристроями; BLE зазвичай працюють на акумуляторі, що є більш гнучким і простішим, ніж Wi-Fi.

Основними складовими даної мережі Bluetooth SIG (Special Interest Group) є [15-16]:

- надійність;
- масштабованість;
- безпека.

В будь-якій системі позиціонування визначення координат мобільного об'єкта проводиться за допомогою заздалегідь розгорнутої інфраструктури – мережі засобів вимірювань, координати яких заздалегідь відомі.

Bluetooth-mesh дозволяє тисячам пристроїв взаємодіяти один з одним при цьому задовольняючи комерційні потреби, а саме масштабованість систем. В специфікації зазначена підтримка до 32000 вузлів мережі. Швидкість передачі даних забезпечується за рахунок малих розмірів пакета Bluetooth разом з високошвидкісним радіо-каналом. Багатоадресна розсилка забезпечується архітектурою ретрансляції керованого потоку повідомлень в поєднанні з можливістю підписки / надання групового обміну повідомленням робить мережу унікальною для обробки великого обсягу даних багатоадресної передачі повідомлень [15].

3 ТЕХНОЛОГІЯ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Концепція технології інтернету речей

Інтернет речей – концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами, за допомогою використання стандартних протоколів зв'язку. Окрім датчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів [17].

Набуває поширення також термін Internet of Everything, IoE – всеохопний, або всеосяжний інтернет. Це явище спричинило занепокоєння в конфіденційності інформації й сприяло появі нового терміну безпека інтернету речей [17].

Основною концепцією IoT є можливість підключення всіляких об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад, холодильник, кондиціонер, автомобіль, велосипед. Всі ці об'єкти (речі) повинні бути оснащені вбудованими давачами або сенсорами, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації. Прикладом впровадження такої концепції є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. У зимовий період регулюються інтенсивність опалення, а в разі спекотної погоди будинок має механізми відкривання і закривання вікон, завдяки чому провітрюється будинок, і все це відбувається без втручання людини [17].

3.2 Технології для побудови систем інтернету речей

Для об'єднання повсякденних речей у мережу потрібні декілька технологій:

- для ідентифікації кожного об'єкту потрібна проста, компактна технологія. Тільки при наявності системи унікальної ідентифікації можна збирати та накопичувати інформацію про певний предмет. Для визначення точного місця знаходження речі можна використовувати технології GPS, або ж локального позиціонування [17];
- для відслідковування змін у стані елементу чи оточуючого середовища об'єкти повинні оснащуватися сенсорами;
- для обробки та накопичення даних з сенсорів повинен використовуватися вбудований комп'ютер (наприклад Raspberry Pi, Intel Edison);
- для обміну інформацією між пристроями можуть бути використані технології бездротових мереж (Wi-Fi, Bluetooth, ZigBee, 6LoWPAN) [17].

3.3 Транспортний засіб до всього (V2X)

Транспортний засіб до всього (V2X) - це зв'язок між транспортним засобом та будь-якою суттю, яка може вплинути на транспортний засіб або може зазнати впливу на нього. Це транспортна система зв'язку, яка включає інші більш специфічні типи зв'язку, як V2I (транспортний засіб до інфраструктури), V2N (транспортний засіб до мережі), V2V (транспортний засіб до транспортного засобу), V2P (транспортний засіб до пішохода) , V2D (транспортний засіб до пристрою) та V2G (транспортний засіб до мережі) [18].

Основними мотивами V2X є безпека дорожнього руху, ефективність руху та економія енергії. Існує два типи технології зв'язку V2X, залежно від використовуваної базової технології:

- WLAN;
- стільниковий зв'язок.

Виділені комунікації короткого радіусу дії (DSRC) - це односторонні або

двосторонні канали бездротового зв'язку від короткого до середнього радіусу дії, спеціально розроблені для автомобільного використання та відповідний набір протоколів та стандартів [19].

Автомобільні системи зв'язку – це комп'ютерні мережі, в яких транспортні засоби та придорожні блоки є вузлами зв'язку, що забезпечують один одного інформацією, такою як попередження про безпеку та інформація про дорожній рух. Вони можуть бути ефективними для уникнення аварій та заторів. Обидва типи вузлів є спеціальними пристроями зв'язку короткого діапазону (DSRC). DSRC працює в діапазоні 5,9 ГГц з пропускну здатністю 75 МГц та приблизним діапазоном 300 м. Автомобільний зв'язок зазвичай розробляється як частина інтелектуальних транспортних систем (ITS) [19].

3.4 Проблеми безпеки

Інтернет речей може викликати величезні зміни у повсякденному житті, надавши звичайним користувачам абсолютно новий рівень комфорту. Але якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, за допомогою надійного криптографічного алгоритму, замість користі вони принесуть шкоду, надавши кібер-злочинцям лазівку для підриву інформаційної безпеки. Оскільки речі із вбудованими комп'ютерами зберігають дуже багато інформації про свого власника, зокрема можуть знати його точне місцезнаходження, доступ до такої інформації може допомогти зловмисникам вчинити злочин [17].

Bluetooth-mesh використовує архітектуру безпеки, яка призначена для вирішення проблем безпеки компаній, що розгортають широкомасштабні мережі бездротових пристроїв.

Рівень управління: пристрої, що додаються в мережу, можуть використовувати алгоритми 256-бітної еліптичної криптографії із застосуванням out-of-band перевірки автентичності. Весь зв'язок в мережі захищений алгоритмом шифрування AES-CCM з 128-бітним ключем. Таким чином гарантується, що всі повідомлення в мережі зашифровані і пройшли перевірку автентичності.

Багаторівневність: шифрування і перевірка справжності реалізуються на двох рівнях: мережевому і прикладному. Всі вузли в мережі можуть ретранслювати повідомлення на мережевому рівні, не знаючи його вміст. Для забезпечення повної безпеки при доставці повідомлення від відправника до одержувача, вміст повідомлення додатково шифрується окремим ключем.

Конфіденційність: кожен пакет мережі обробляється так, щоб видалити будь-яку ідентифікаційну інформацію. Це не дозволяє іншим відстежувати пристрої мережі, особливо коли ці пристрої переміщуються в зоні дії інших мереж [15].

3.5 Висновки до розділу

Як відомо, надійність мережі оцінюється в тому числі і по її здатності доставляти повідомлення від одного пристрою до іншого. Для забезпечення безперервної доставки повідомлень мережа Bluetooth-mesh використовує два типи зв'язку:

- Peer-to-Peer: однорангова мережа, описана в специфікації технології яка називається «piconet», завдяки якій, всі вузли мережі можуть з'єднуватися один з одним безпосередньо без додаткових спеціальних вузлів з функцією концентратора або маршрутизатора. Що в свою чергу позбавляє мережу від точок відмови;

- Multipath (message relay): архітектура ретрансляції керованого потоку повідомлень з використанням декількох маршрутів. У зв'язці з простотою розгортання і управління є надійним способом доставки повідомлень[15].

Саме тому в даній роботі було вирішено використовувати мережу Bluetooth-mesh, як спосіб об'єднання давачів (маяків) для локального позиціонування в одне ціле з системою. Що допоможе вирішити поставлені задачі по типу ідентифікації об'єктів в мережі та забезпечення безпеки в передачі даних.

4 РОЗРОБЛЕННЯ СИСТЕМИ АВТОМАТИЧНОГО ПАРКУВАННЯ

4.1 Розроблення та опис структурної схеми

Виходячи з технічного завдання і результатів аналізу подібних рішень в роботі була розроблена структурна схема системи автоматичного паркування автомобілів на базі технології інтернету речей, яка представлена на кресленику (Додаток А). Структурна схема системи призначена для відображення загальної структури системи, а саме її основних блоків, підсистем та головних зв'язків між ними.

Система розділена на дві підсистеми, а саме:

- підсистема автомобіля;
- підсистема парковки.

В свою чергу підсистема автомобіля розділена на чотири групи:

- блок давачів та інформаційні пристрої (ВК);
- блок виконавчих пристроїв автомобіля;
- блок безпроводних модулів автомобіля;
- пристрої обробки інформації (бортовий комп'ютер та інші блоки керування).

До першої групи відносяться камери, лідар(и), ультразвукові давачі, радары, давачі тиску гальмівної системи, давач кута повороту керма. Дані, що надходять з них передаються далі для подальшої обробки.

Пристрої обробки інформації, до яких входить бортовий комп'ютер та блоки керування виконавчих пристроїв. Результатом роботи бортового комп'ютера є обробка даних з давачів, камер та надсилання потоку даних до підсистеми парковки через блок безпроводних модулів, та отримання від системи команд керування для оброблення та передачі до блоків керування відповідних виконавчих пристроїв.

Підсистема парковки розділена на чотири групи:

- блок давачів парковки та інформаційних підсистем (ППБ, ПВС);
- блок виконавчих пристроїв парковки;
- блок безпроводних модулів парковки;

- пристрої обробки інформації(блоки керування, сервери).

Дані з автомобіля потрапляють через модулі зв'язку (безпроводні модулі, Ethernet) до серверу даних. Водночас данні з давачів парковки: давачів руху, давачів освітлення коли це потрібно, але в основному це данні про зайняті місця та данні локалізації з таких виконавчих пристроїв як блок маяків (міток) передаються до серверу сервісів. Особливість такої структури полягає в тому, що мітки самі по собі не є елементами інфраструктури. Тому вони не беруть участь у формуванні мережі, маршрутизації тощо. Тобто звільнені від всіх завдань, крім вимірювання відстаней до мобільних об'єктів (автомобілів) і передачі результатів на сервер через транспортну інфраструктуру. Це дозволяє істотно спростити мітки і відповідно знизити енергоспоживання і вартість міток. Мітки взаємодіють з інфраструктурою через двонаправлений радіо інтерфейс. Інтерфейс визначає частотний діапазон, форму радіосигналу, способи модуляції і кодування, формати пакетів, а також команди та звіти, якими обмінюються мітки і інфраструктура.

Весь так званий цикл передачі даних між елементами системи можна побачити в (додатку Д) діаграма послідовностей системи.

Діаграму умовно можна розділити на складові елементи:

- 1) авто (певний додаток встановлений на бортовий комп'ютер автомобіля) через модулі зв'язку створює запит на підключення автомобіля до системи (передача певного ідентифікатора для авторизації користувача);
- 2) встановлення каналу зв'язку з системою, перевірка клієнта в базі даних (дозвіл на резервування медіа ресурсу);
- 3) запит на виділення медіа ресурсу (повернення певного IP), на який в дальнішому будуть відправлятися медіа дані з автомобіля;
- 4) збір даних з давачів та відео камер авто, формування медіа потоку та надсилання до виділеного ресурсу;
- 5) водночас відбувається збір даних з давачів паркінгу та їх передача разом з даними авто до серверу сервісів для розрахунку контрольних точок, маршруту та маневру паркування;

- 6) після чого всі розрахунки використовуються сервером додатків для формування команд керування;
- 7) блок керування рухом авто (БКР) надсилає команди до бортового комп'ютера автомобіля;
- 8) команди попередньо опрацьовуються та поступають на відповідні блоки керування потрібних виконавчих пристроїв.

Для кращого розуміння системи розглянемо (додаток Г) – основні етапи функціонування системи. Оскільки для простоти реалізації даної системи створення бази цифрових відбитків та динамічне уникнення перешкод виключається та надається модель середовища у вигляді карти довкілля системою транспортного засобу до інфраструктури V2X або камерами паркінгу. Тобто система використовує статичну карту паркінгу і припускає, що самолокалізація автомобіля є точною.

Після підключення автомобіля до системи відбувається визначення вихідної точки (положення) автомобіля на карті для подальшого планування маршруту. Маршрут створюється з допомогою карти та статичного глобального плану маршруту, що по суті відображає цілі у вигляді певних координат на карті. Тому запит (перевірка) місця знаходження буде тривати доти поки автомобіль не досягне кінцевої цілі.

Після розбору та перегляду функціональної схеми, етапів функціонування системи та діаграми послідовностей перейдемо до діаграми прецедентів, зображеної в (додатку Е) щоб розібратись як відбувається рух авто.

Після знаходження перехідних положень (цілей) з запланованого шляху, та маневру паркування на сервері сервісів данні зберігаються чи оновлюються на сервері даних, після чого використовуються на сервері додатків для формування команд керування. Блок керування рухом авто (БКР) через модулі зв'язку надсилає їх до автомобіля. Після чого відбувається попередня обробка команд керування бортовим комп'ютером автомобіля та передача відповідним блокам керування виконавчих пристроїв. Наприклад профіль швидкості, команди прискорення та уповільнення опрацьовує блок електродвигунів (БЕ) та блок гальмівних приводів

(БГП) для скидання швидкості при зміні напрямку або зупинки в кінцевій траєкторії.

4.2 Розроблення та опис функціональної схеми

Схема електрична функціональна – схема, що роз'яснює певні процеси, які протікають в окремих функціональних вузлах виробу або у виробі в цілому.

Розроблена функціональна схема представлена на креслениках (додаток Б), (додаток В).

На схемі (додаток Б) зображено функціональні особливості елементів підсистеми автомобіля.

Наприклад, після отримання сигналу з світлочутливих матриць відеокамери, інформація за допомогою аналого-цифрових перетворювачів оцифровується та надходить до блоків цифрової обробки, де відбувається попередня обробка зображень. Далі, через пристрої прийому-передачі мережі MOST відбувається зв'язок з блоком управління або бортовим комп'ютером.

Структура лідару схожа, але має певні особливості пов'язані з будовою. Лідар потребує підсилювача після світло чутливої матриці. Далі інформація так само перетворюється, обробляється і передається до блоку керування та бортового комп'ютера.

Радари та ультразвукові давачі відрізняються типом підключення; для перших застосовується CAN-шина, а для других – мережа MOST. Передавальний пристрій (чутливий елемент) посилає ультразвуковий імпульс, який після відбиття повертається назад, відбитий сигнал підсилюється, та за допомогою аналого-цифрових перетворювачів оцифровується та надходить до блоків цифрової обробки обробляється і передається до блоку керування та бортового комп'ютера .

Модулі безпроводного зв'язку зображені як пристрої прийому-передачі підключені через SPI-шину до блоку керування.

Розглянемо функціональну схему підсистеми парковки.

На схемі (додаток В) зображено функціональні елементи підсистеми парковки.

Модулі безпроводного зв'язку зображені як пристрої прийому-передачі, підключені через SPI-шину до БКР.

Давачі зайнятості місць мають наступну структуру: передавальний пристрій (чутливий елемент) посиляє ультразвуковий імпульс, який після відбиття повертається назад, відбитий сигнал підсилюється, та за допомогою аналого-цифрових перетворювачів оцифровується та надходить до блоків цифрової обробки обробляється і передається до блока керування (БК) через CAN-шину.

Давачі освітлення мають наступну функціональну структуру: світлочутливі матриці та детектор, що після досягнення певного рівня яскравості створює певну напругу, яка за допомогою аналого-цифрових перетворювачів оцифровується та надходить до блоків цифрової обробки. Далі, через пристрої прийому-передачі через мережу MOST відбувається зв'язок з блоком управління.

У якості Bluetooth модулів для маяків було обрано модулі Bluetooth 5.0 NRF52840 основні технічні характеристики наведені у таблиці 4.1. У якості методу кодування сигналу використовується метод GFSK (Gaussian Frequency-Shift Keying), який дозволяє збільшити показники завадостійкості [22-24].

Однією з особливостей даного модулю є досить низьке енергоспоживання, що дозволяє використовувати його в переносних пристроях з батарейним живленням, що і потрібно для Bluetooth маяків. Споживання струму модулем становить 35 мА і 100 мкА – в «сплячому» режимі. Напруга може змінюватися в межах від 1,7 до 3,6 В [22-24].

Узгоджувальний каскад дозволяє забезпечувати максимальну передачу вихідної потужності приймача антени шляхом узгодження вихідного опору приймача і вхідного опору антени. Смуговий фільтр пропускає сигнали в певному діапазоні частот, і послаблює сигнали що виходять за вказаний діапазон.

Крім аналогової частини в модулях, пристрої можуть різнитись в залежності від виробника і в цифрових вузлах: MAC-контролер, SPI, дисперсійну лінію затримки (DDDL) т.і. MAC-контролер реалізує два способи доступу до оточення:

- множинний доступ з контролем несучої і запобіганням колізій;
- множинний доступ з тимчасовим поділом каналів.

SPI здійснює зв'язок приймача з контролером, або за допомогою SPI Ethernet модуля наприклад (ENC28J60) додати роз'єм Ethernet [25].

Таблиця 4.1 – Основні технічні характеристики NRF52840 Bluetooth 5.0 [22]

Найменування параметру	Значення
Частотний діапазон	2,40-2,48 ГГц; не ліцензований
Стандарт радіозв'язку	Bluetooth 5
Смуга частот	80 МГц, 22 МГц
Метод кодування сигналу	Гаусівська-частотна модуляція (GFSK)
Бітова швидкість передачі даних	2 Мбіт/с
Точність вимірювання відстаней	До 2 метрів
Метод вимірювання відстаней	TOF, RSSI
Напруга живлення	1,7 ... 3,6 В
Струм споживання передавача	35 мА
Струм споживання передавача в режимі сну	100 мкА
Температурний діапазон	- 30 °C ... + 77 °C

5 МОДЕЛЮВАННЯ СИСТЕМИ АВТОМАТИЧНОГО ПАРКУВАННЯ

5.1 Огляд системи автоматичного паркування

Автоматична парковка автомобіля, є складною проблемою. Очікується, що автоматизовані системи транспортного засобу заберуть контроль і направлять транспортний засіб на доступне місце для паркування. Така функція використовує безліч бортових датчиків. Наприклад:

- передні та бічні камери для виявлення розмітки смуги руху, дорожніх знаків (знаки зупинки, розмітки виїзду тощо), інших транспортних засобів та пішоходів.
- лідар та ультразвукові датчики для виявлення перешкод та розрахунку точних вимірювань відстані.
- ультразвукові датчики для виявлення перешкод.
- IMU та колісні кодери для розрахування мертвих зон.

Вбудовані датчики використовуються для сприйняття навколишнього середовища навколо автомобіля. Сприйняте середовище включає розуміння дорожньої розмітки для тлумачення правил дорожнього руху та виведення ділянок, за якими можна пересуватися, розпізнавання перешкод та виявлення доступних місць для паркування.

Оскільки датчики транспортного засобу сприймають світ, транспортний засіб повинен спланувати шлях через навколишнє середовище до вільного місця для паркування та виконати послідовність дій, необхідних для проїзду до нього. Роблячи це, він повинен реагувати на динамічні зміни навколишнього середовища, такі як пішоходи, які перетинають його шлях, та коригувати свій план.

В роботі було реалізовано підмножину функцій, необхідних для реалізації такої системи. Вона зосереджена на плануванні здійсненого шляху в навколишньому середовищі та здійсненні дій, необхідних для проходження шляху.

5.2 Опис моделі MATLAB Simulink

Перед моделюванням функція `helperSLCreateCostmap` викликається в рамках функції зворотного виклику моделі `PreLoadFcn`. Функція `helperSLCreateCostmap` створює статичну карту на стоянці, яка містить інформацію про стаціонарні перешкоди, дорожню розмітку та припаркованих автомобілях. Карта представлена у вигляді об'єкта `vehicleCostmap`.

Для використання об'єкта `vehicleCostmap` в Simulink функція `helperSLCreateUtilityStruct` перетворює `vehicleCostmap` у масив структури в ініціалізації маски блоку. Результат використання об'єкта `vehicleCostmap` на рисунку 5.1.

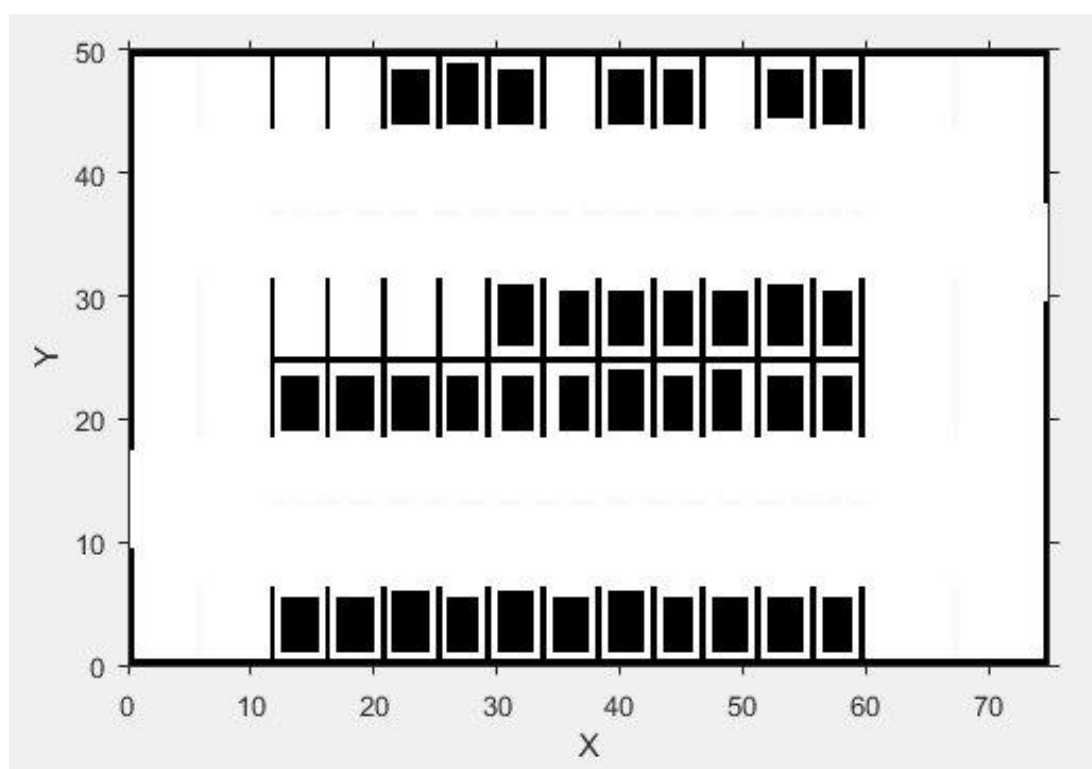


Рисунок 5.1 – Карта парковки

Глобальний план маршруту описується як послідовність відрізків смуги, яку потрібно пройти, щоб дістатися до місця стоянки. Перед моделюванням функція `PreLoadFcn` зворотного виклику моделі завантажує план маршруту, який

зберігається у вигляді таблиці. В таблиці 5.1 вказані початкові та кінцеві позиції сегмента, а також властивості сегмента, такі як обмеження швидкості.

Таблиця 5.1 – План маршруту

StartPose	EndPose	Attributes
4;12;0	56;11;0	[1×1 struct]
56;11;0	70;19;90	[1×1 struct]
70;19;90	70;32;90	[1×1 struct]
70;32;90	52,38,180	[1×1 struct]
53;39;180	36,3; 44; 90	[1×1 struct]

Вхідними та вихідними даними багатьох блоків у моделі (додаток Ж) є шини Simulink (класи Simulink.Bus (Simulink)). У функції PreLoadFcn зворотного виклику моделі ці шини створює функція helperSLCreateUtilityBus.

Планування – це ієрархічний процес, кожен наступний шар відповідає за більш дрібне завдання. Шар поведінки знаходиться у верхній частині цього стека. Блок планувальник поведінки (Behavior Planner) запускає послідовність навігаційних завдань на основі глобального плану маршруту, надаючи проміжні цілі та конфігурацію для блоку планування руху (Path Planner) та формування траєкторії. Транспортний засіб здійснює навігацію по кожному сегменту шляху, виконуючи такі дії:

- 1) планування руху: планування можливого шляху через карту, використовуючи оптимальний алгоритм швидкого вивчення випадкового дерева RRT* за допомогою Path Planner;
- 2) згладжування контуру: згладжуємо опорний шлях, пристосувавши до нього сплайни використовуючи Path Smoothing Spline;
- 3) генерація траєкторії: перетворюємо згладжений шлях у траєкторію, створивши профіль швидкості за допомогою Speed Profile Generator;

4) керування транспортним засобом: враховуючи згладжений контрольний шлях, Path Analyzer обчислює еталонну позицію та швидкість на основі поточної позиції та швидкості автомобіля. Наданий опорними значеннями, lateral Controller обчислює кут повороту для контролю напрямку руху автомобіля. Longitudinal Controller обчислює команди прискорення та уповільнення для підтримання бажаної швидкості автомобіля.

Перевірка цілей: перевіряємо, чи транспортний засіб досягнув кінцевої цілі за допомогою Goal Checker. Цілі зображено на рисунку 5.2.

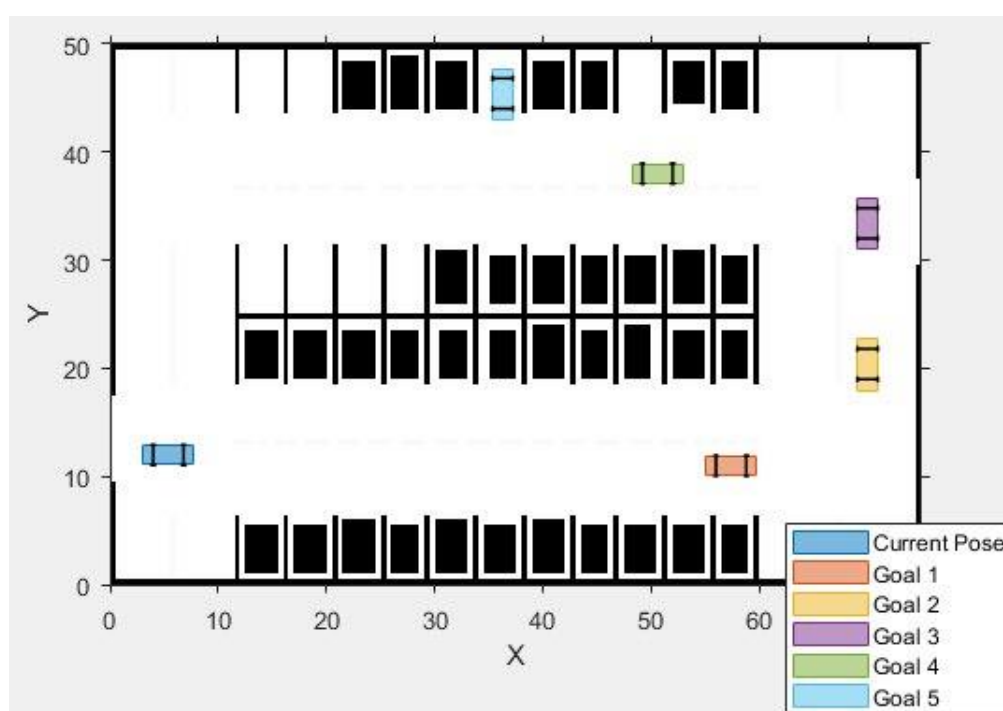


Рисунок 5.2 – Послідовність цілей до місця стоянки

5.2.1 Підсистема контролера автомобіля

Підсистема "Контролер транспортного засобу", що зображена у (додатку 3) та рисунку 5.3 містить блок бічного контролера Stanley та поздовжнього контролера Stanley для регулювання пози та швидкості руху автомобіля відповідно. Для обробки реалістичної динаміки транспортного засобу, параметр моделі автомобіля в блоці бічного контролера Stanley встановлений на Dynamic bicycle model. З цією конфігурацією для обчислення команди рульового управління потрібні додаткові

входи, такі як кривизна шляху, поточна швидкість похилу транспортного засобу та поточний кут повороту. Блок поздовжнього контролера Stanley використовує перемикаючий пропорційно-інтегральний (ПІ) контролер для розрахунку команд прискорення та уповільнення, які приводять в дію гальмо та дросель у транспортному засобі.

5.2.2 Блок поздовжнього контролера

Блок поздовжнього контролера Stanley обчислює команди прискорення та уповільнення в метрах за секунду, які контролюють швидкість автомобіля. Для цього на вхід подається еталонна швидкість, поточна швидкість та поточний напрямок руху. Контролер обчислює ці команди за допомогою методу Stanley, який блок реалізує як дискретний ПІ контролер з методом «Anti-windup».

Обчислення команд кута повороту автомобіля також за допомогою методу Stanley, у блоці бічного контролера.

Вхідні дані поздовжнього контролера:

- 1) RefVelocity – референтна швидкість. Опорна швидкість, в метрах за секунду, вказана як дійсний скаляр;
- 2) CurrVelocity – справжня шкала поточної швидкості. Поточна швидкість транспортного засобу, в метрах на секунду, вказана як дійсний скаляр;
- 3) Direction – напрямок руху транспортного засобу, зазначений як 1 (рух вперед) -1 (зворотний рух);
- 4) Reset – тригер для скидання інтегралу похибки швидкості 0 (утримувати стабільно) ненульовий скаляр (скидання) тригер для скидання інтегралу похибки швидкості, $e(k)$, до нуля. Значення 0 утримує $e(k)$ стабільно. Ненульове значення скидає $e(k)$.

Вихідні дані поздовжнього контролера:

- 1) AccelCmd – команда прискорення, повернена як дійсний скаляр в діапазоні $[0, M_A]$, де M_A – значення параметра максимальне поздовжнє прискорення (м/с^2);

2) DecelCmd – команда уповільнення, повернена як дійсний скаляр в діапазоні $[0, M_D]$, де M_D – значення параметра максимальне поздовжнє уповільнення (м/с^2).

Алгоритми

Блок поздовжнього контролера Stanley реалізує дискретний ПІ контролер з інтегрованою системою проти обриву інтегратора, коли вихідний сигнал насичений, якій описано параметром методу Anti-windup (Simulink) блоку ПІД контролер. Цей контролер перевіряє вихід на насичення, і якщо члени Р+І перевищують бажаний діапазон регулювання, то нове значення інтегратора не зберігається. Блок використовує таке рівняння:

$$u(k) = \left(K_p + K_i \frac{T_s z}{z-1} \right) e(k), \text{ де}$$

$u(k)$ – сигнал управління на k -му кроці часу;

K_p – пропорційний коефіцієнт посилення, встановлений параметром пропорційного коефіцієнта посилення, K_p ;

K_i – інтегральний коефіцієнт посилення, встановлений параметром інтегрального коефіцієнта посилення, K_i ;

T_s – час вибірки блоку в секундах, встановлений параметром час вибірки (с);

$e(k)$ – похибка швидкості ($\text{CurrVelocity} - \text{RefVelocity}$) на k -му кроці часу. Для кожного k ця помилка дорівнює різниці між поточною швидкістю та вхідними значеннями швидкості ($\text{CurrVelocity} - \text{RefVelocity}$).

Сигнал управління, u , визначає значення команди прискорення AccelCmd і команди уповільнення DecelCmd. Блок насичує команди прискорення та уповільнення до відповідних діапазонів $[0, M_A]$, та $[0, M_D]$, де:

M_A – значення параметра максимального поздовжнього прискорення $\frac{\text{м}}{\text{с}^2}$;

M_D – значення параметра максимального поздовжнього уповільнення $\frac{\text{м}}{\text{с}^2}$.

На кожному кроці часу лише одне із значень портів AccelCmd та DecelCmd є позитивним, а інше значення порту – 0. Іншими словами, транспортний засіб може або прискорюватися, або гальмувати за один часовий крок, але не може робити і те і інше одночасно.

Напрямок руху, як зазначено у вхідному порту Direction, визначає, яка команда є позитивною на даному кроці часу. Значення порту Direction вказані в таблиці 5.2.

Таблиця – 5.2 Значення порту Direction

Значення порту напрямку	Значення керуючого сигналу $u(k)$	Значення порту AccelCmd	Значення порту DecelCmd	Опис
	$u(k) > 0$	позитивний дійсний скаляр	0	Автомобіль пришвидшується, рухаючись вперед
	$u(k) < 0$	0	позитивний дійсний скаляр	Автомобіль уповільнює рух, рухаючись вперед
	$u(k) > 0$	0	позитивний дійсний скаляр	Автомобіль уповільнює рух під час руху заднім ходом
	$u(k) < 0$	позитивний дійсний скаляр	0	Транспортний засіб пришвидшується, коли рухається у зворотному напрямку

5.2.3 Блок бічного контролера

Блок бічний контролера Stanley, що зображено на (додатку Ж) та рисунку 5.3 обчислює команду кута повороту в градусах, яка регулює поточну позицію автомобіля відповідно до опорної позиції, враховуючи поточну швидкість та напрямок руху автомобіля. Контролер обчислює цю команду методом Stanley, закон управління якого базується як на кінематичній, так і на динамічній моделі велосипеда. Для перемикання між моделями використовується параметр моделі автомобіля. Перемикання залежить від умов модуляції руху:

- кінематична модель велосипеда підходить для проходження траси в низько швидкісних середовищах, таких як стоянки, де інерційні ефекти мінімальні;
 - динамічна модель велосипеда підходить для руху по трасі у високошвидкісних середовищах, таких як шосе, де інерційні ефекти більш виражені.
- Ця модель транспортного засобу забезпечує додаткові параметри, що описують динаміку автомобіля.

Вхідні дані:

- RefPose – опорна (еталонна) позиція $[x, y, \theta]$. Вектор опорної позиції, зазначений як вектор $[x, y, \theta]$, x та y – в метрах, а θ – у градусах. x та y вказують контрольну точку для керування транспортним засобом. θ визначає кут орієнтації шляху в цій опорній точці і є додатним у напрямку проти годинникової стрілки. Для транспортного засобу, що рухається вперед, опорною точкою є точка на шляху, найближча до центру передньої осі транспортного засобу. Для транспортного засобу, що рухається у зворотному напрямку, опорною точкою є точка на шляху, що знаходиться найближче до центру задньої осі транспортного засобу;
- CurrPose – поточна позиція $[x, y, \theta]$. Вектор поточної позиції транспортного засобу, зазначений як вектор $[x, y, \theta]$. x та y – в метрах, а θ – у градусах. x та y визначають місце розташування транспортного засобу, яке визначається як центр задньої осі транспортного засобу. θ визначає кут орієнтації

транспортного засобу в місці (x, y) і додатний у напрямку проти годинникової стрілки;

- CurrVelocity – дійсний скаляр поздовжньої швидкості. Поточна поздовжня швидкість транспортного засобу, вказана як дійсний скаляр. Одиниці вимірювання – в метрах на секунду. Якщо транспортний засіб рухається вперед, це значення має бути більше 0. Якщо транспортний засіб рухається заднім ходом, це значення має бути менше 0. Значення 0 означає, що транспортний засіб не рухається;

- Direction – напрямок руху транспортного засобу, зазначений як 1 для руху вперед, так і -1 для зворотного руху. Напрямок руху визначає похибку положення та похибку кута, що використовуються для обчислення команди кута повороту;

- Curvature – кривизна шляху в опорній точці, в радіанах на метр, вказана як дійсний скаляр. Для транспортного засобу, що рухається вперед, опорною точкою є точка на шляху, найближча до центру передньої осі транспортного засобу. Для транспортного засобу, що рухається у зворотному напрямку, опорною точкою є точка на шляху, що знаходиться найближче до центру задньої осі транспортного засобу. отримати кривизну шляху можна від вихідного порту Curvatures блоку Path Smoothing Spline, а також отримати кривизни меж смуги від вихідного порту Curvature блоку Path Analyzer;

- CurrYawRate – поточна швидкість відхилення транспортного засобу, в градусах на секунду, вказана як дійсний скаляр. Поточна швидкість відхилення – це швидкість зміни в кутовій швидкості автомобіля;

- CurrSteer – поточний кут повороту. Поточний кут повороту автомобіля, в градусах, вказаний як дійсний скаляр. Це значення додатне у напрямку проти годинникової стрілки.

Вихідні дані:

- SteerCmd – команда кута повороту в градусах повернута як дійсний скаляр. Це значення додатне у напрямку проти годинникової стрілки.

Для обчислення команди кута повороту контролер мінімізує похибку положення та похибку кута поточної позиції щодо опорної позиції. Напрямок руху автомобіля визначає ці значення похибки.

Коли автомобіль рухається вперед (параметр напрямку (Direction) – 1):

- помилка положення є бічною відстанню від центру передньої осі до опорної точки на шляху;
- помилка кута є кут переднього колеса щодо опорної траєкторії.

Коли транспортний засіб перебуває в реверсному русі (параметр напрямку (Direction) – -1):

- помилка положення є бічною відстанню від центру задньої осі до опорної точки на шляху;
- помилка кута є кут заднього колеса щодо опорної траєкторії.

Кінематична модель: Кінематична модель передбачає, що транспортний засіб має незначну інерцію. Ця конфігурація в основному підходить для низько швидкісних середовищ, де інерційні ефекти мінімальні. Команда керування обчислюється на основі опорної пози, поточної пози та швидкості руху автомобіля.

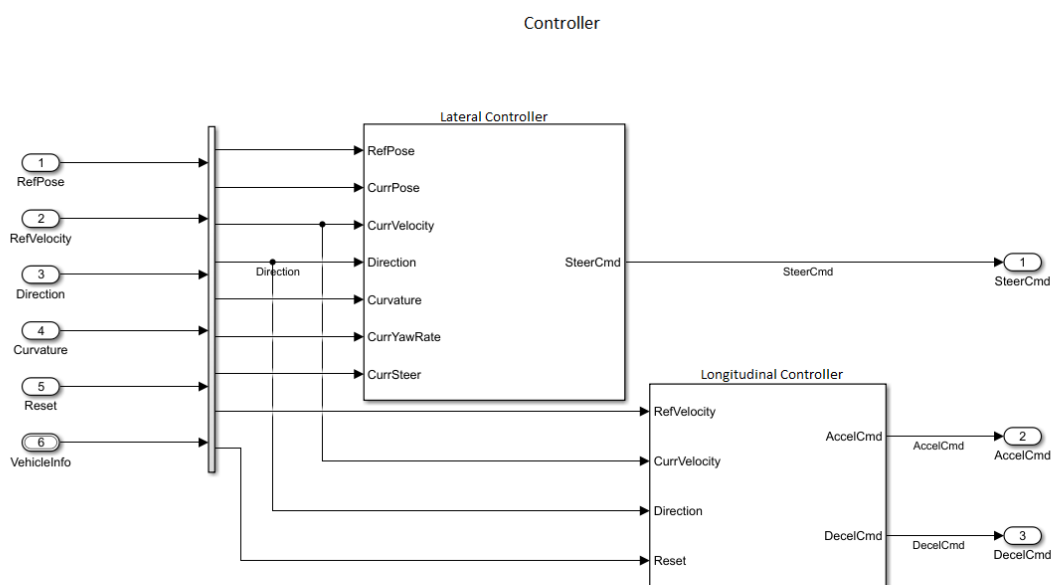


Рисунок 5.3 – Контролер моделі автомобіля

5.2.4 Модель авто

Щоб продемонструвати ефективність, контролер транспортного засобу застосовується до блоку Car Model, який містить спрощену систему рульового управління, яка змодельована як система першого порядку, та блок Vehicle Body 3DOF (Vehicle Dynamics Blockset), спільно використовуваний між Automated Driving Toolbox та Vehicle Dynamics Blockset. Модель враховує інерційні ефекти, такі як ковзання шини та спрацьовування рульового сервоприводу.

Блок Vehicle Body 3DOF реалізує жорстку двовісну модель кузова автомобіля для обчислення поздовжнього, поперечного та поворотного руху. Блок враховує масу тіла та аеродинамічний опір між осями через прискорення та рульове управління.

У бібліотеці Vehicle Dynamics Blockset є два типи блоків 3DOF Body, які моделюють поздовжній, поперечний та поворотний рух.

Налаштування колії автомобіля (Single line) Сили діють уздовж центральної лінії на передній і задній осях.

Вхідний порт WhlAngF кут повороту переднього колеса, δ_F , в рад. Розмір сигналу скаляр – 1.

Вхідний порт xdotin – поздовжня швидкість передана як скаляр.

Швидкість CG (Centr of Gravity) центра сили тяжіння автомобіля по фіксованій осі x, зафіксованій на транспортному засобі, м/с.

Вихідні порти:

psi – відхилення, повернуте як скаляр. Обертання рами, закріпленої на автомобілі, щодо зафіксованої на землі осі Z (відхилення), в рад;

r – швидкість відхилення, повернуте як скаляр. Кутова швидкість автомобіля, r, щодо зафіксованої на транспортному засобі осі z (швидкість відхилення), в рад/с;

xdot – поздовжня швидкість транспортного засобу, повернуте як скаляр. Центр сили тяжіння транспортного засобу вздовж закріпленої на транспортному засобі осі x, м/с;

$ydot$ – поперечна швидкість транспортного засобу, повернуте як скаляр. Центр сили тяжіння транспортного засобу по фіксованій осі y , м/с;

Info – сигнал шини, що містить значення блоків displacement (Переміщення CG транспортного засобу вздовж зафіксованих на землі X,Y,Z – осей, обчислюється в метрах), velocity (Швидкість руху транспортного засобу вздовж зафіксованої на землі XYZ – осі обчислюється в метрах за секунду), Ang (Кут Обертання нерухомої рами транспортного засобу навколо зафіксованої на землі Z- осі (відхилення) обчислюється в рад.)

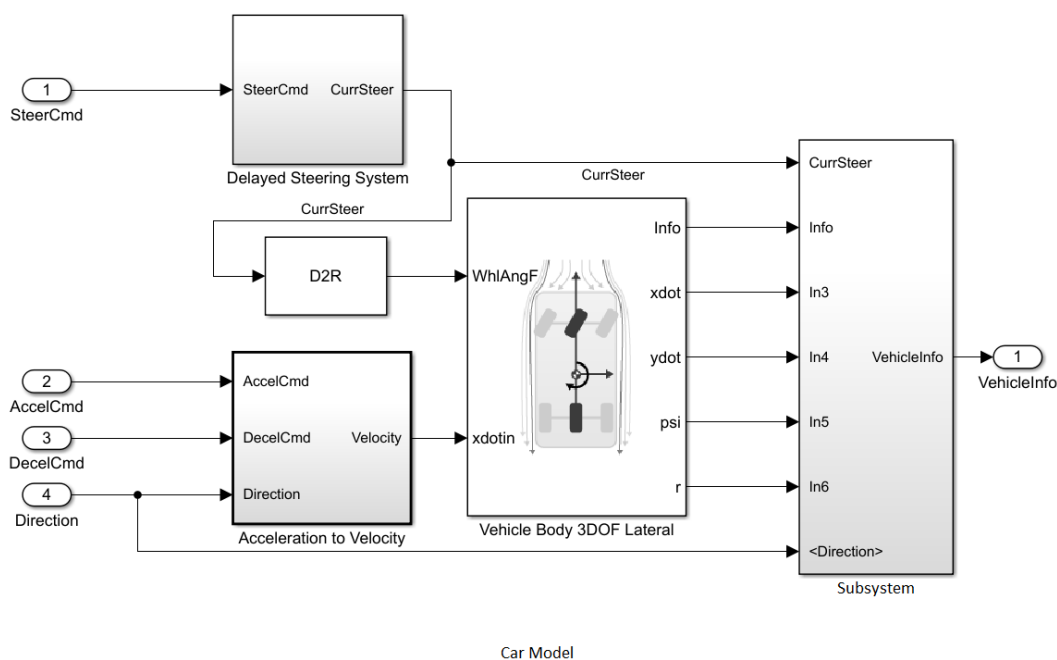


Рисунок 5.4 – Модель атомобіля

Базові швидкості, разом із згладженим шляхом, складають траєкторію руху транспортного засобу. Для дотримання цієї траєкторії використовується контролер зворотного зв'язку. Контролер виправляє помилки при відстеженні траєкторії, що виникають внаслідок ковзання шин та інших джерел, таких як неточності в локалізації. Зокрема, контролер складається з двох компонентів:

- Бічне управління: врегулювання кута повороту таким чином, щоб транспортний засіб рухався по еталонному шляху;

– Поздовжнє управління: Проводячи контрольний шлях, підтримуємо бажану швидкість, керуючи дроселем та гальмом.

Оскільки сценарій передбачає повільні швидкості, було спрощено контролер, враховуючи лише кінематичну модель. Бічне управління реалізується функцією `lateralControllerStanley`. Поздовжнє управління здійснюється за допомогою допоміжного System object `HelperLongitudinalController`, який обчислює команди прискорення та уповільнення на основі ПІ-закону.

Для контролера зворотного зв'язку потрібен симулятор, який може виконувати бажані команди контролера за допомогою відповідної моделі автомобіля. Клас `HelperVehicleSimulator` імітує такий транспортний засіб з допомогою наступної кінематичної моделі:

$$\begin{aligned}\dot{x}_r &= v_r * \cos(\theta), \\ \dot{y}_r &= v_r * \sin(\theta), \\ \dot{\theta} &= \frac{v_r}{l} * \tan(\delta), \\ \dot{v}_r &= a_r.\end{aligned}$$

У наведених рівняннях (x_r, y_r, θ) представляє позицію транспортного засобу у світових координатах. v_r, a_r, l і δ представляють швидкість заднього колеса, прискорення заднього колеса, колісну базу та кут повороту відповідно. Положення та швидкість руху переднього колеса можна отримати за допомогою:

$$\begin{aligned}x_f &= x_r + l \cos(\theta), \\ y_f &= y_r + l \sin(\theta), \\ v_f &= \frac{v_r}{\cos(\delta)}.\end{aligned}$$

Створення моделі автомобіля, та надання йому потрібних властивостей

```
vehicleSim = HelperVehicleSimulator(costmap, vehicleDims);
```

Встановлюємо значення для позиції та швидкості руху автомобіля

```
vehicleSim.setVehiclePose(currentPose);
```

```
currentVel = 0;
```

```
vehicleSim.setVehicleVelocity(currentVel);
```

Налаштовуємо модель для відображення траєкторії

```
vehicleSim.showTrajectory(true);
```

Створюємо об'єкт HelperPathAnalyzer для обчислення опорної позиції, опорної швидкості та напрямку руху для контролера.

```
pathAnalyzer = HelperPathAnalyzer(refPoses, refVelocities, directions, ...
    'Wheelbase', vehicleDims.Wheelbase);
```

Створюємо об'єкт HelperLongitudinalController для контролю швидкості руху автомобіля та вказуємо час вибірки.

```
sampleTime = 0,05;
lonController = HelperLongitudinalController ( 'SampleTime' , sampleTime);
```

Використаємо об'єкт HelperFixedRate, щоб забезпечити виконання контролера зворотного зв'язку з фіксованою швидкістю. Використовуємо швидкість управління, щоб відповідати поздовжньому контролеру.

```
controlRate = HelperFixedRate (1 / sampleTime); // В герцах
```

Поки ціль не досягнута, виконуємо наступне:

- Обчислення команди рульового управління та прискорення / уповільнення, необхідні для відстеження запланованої траєкторії;
- Подаємо команди управління в тренажер;
- Записуємо повернуту позицію транспортного засобу та швидкість передачі в контролер під час наступної ітерації.

```
reachGoal = false;
```

```
while ~reachGoal;
```

Знаходження опорної позиції на шляху та відповідної швидкості

```
[refPose, refVel, direction] = pathAnalyzer(currentPose, currentVel);
```

Оновлення напрямку руху моделі авто

```
updateDrivingDirection(vehicleSim, direction);
```

Обчислення команд керування

```
steeringAngle = lateralControllerStanley(refPose, currentPose, currentVel, ...
    'Direction', direction, 'Wheelbase', vehicleDims.Wheelbase);
```

Обчислення команд прискорення та уповільнення

```
lonController.Direction = direction;
```

```
[accelCmd, decelCmd] = lonController(refVel, currentVel);
```

Імітація транспортного засобу за допомогою виходів контролера

```
drive(vehicleSim, accelCmd, decelCmd, steeringAngle);
```

Перевірка чи досягає транспортний засіб мети

```
reachGoal = helperGoalChecker(goalPose, currentPose, currentVel, endSpeed, direction);
```

Очікування виконання фіксованого положення

```
waitfor(controlRate);
```

Отримання поточної позиції та швидкості руху автомобіля

```
currentPose = getVehiclePose(vehicleSim);
```

```
currentVel = getVehicleVelocity(vehicleSim);
```

```
end;
```

Відображення моделювання руху авто зображено на рисунку 5.5.

```
showFigure(vehicleSim);
```

Це завершує перший етап плану маршруту та демонструє кожен крок процесу.

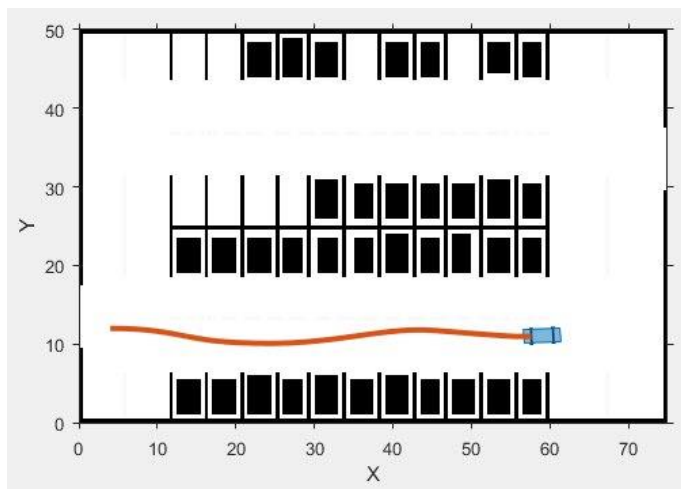


Рисунок 5.5 – Моделювання руху автомобіля

5.2.5 Результати моделювання

Блок візуалізації показує, як транспортний засіб відстежує контрольний шлях. Він також відображає швидкість автомобіля та команди рульового управління в межах. Наступні зображення є результати моделювання, та зображені на рисунках 5.6, 5.7.

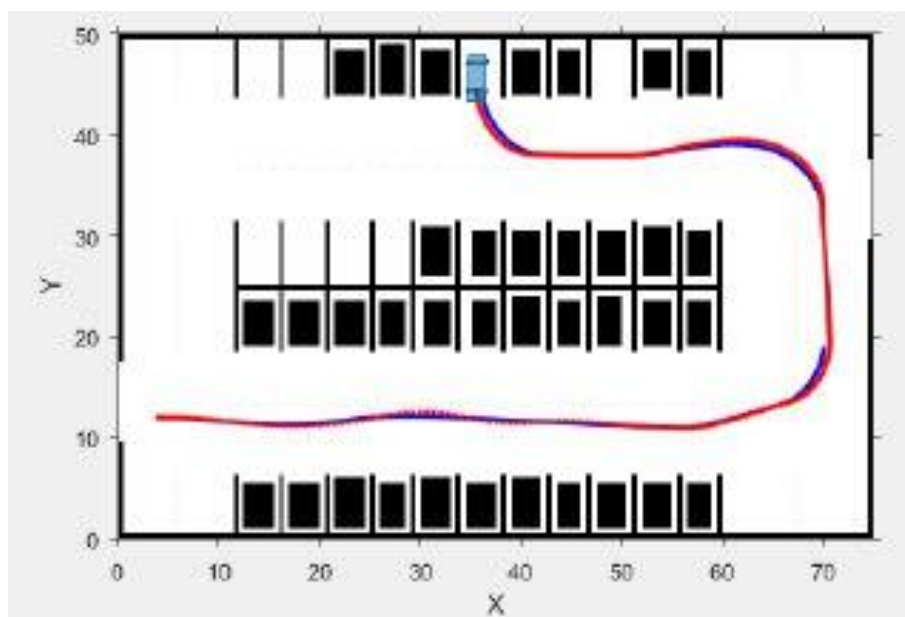


Рисунок 5.6 – Контрольний шлях до місця парковки

Моделювання зупиняється приблизно через 45 секунд, тобто коли транспортний засіб досягає пункту призначення.

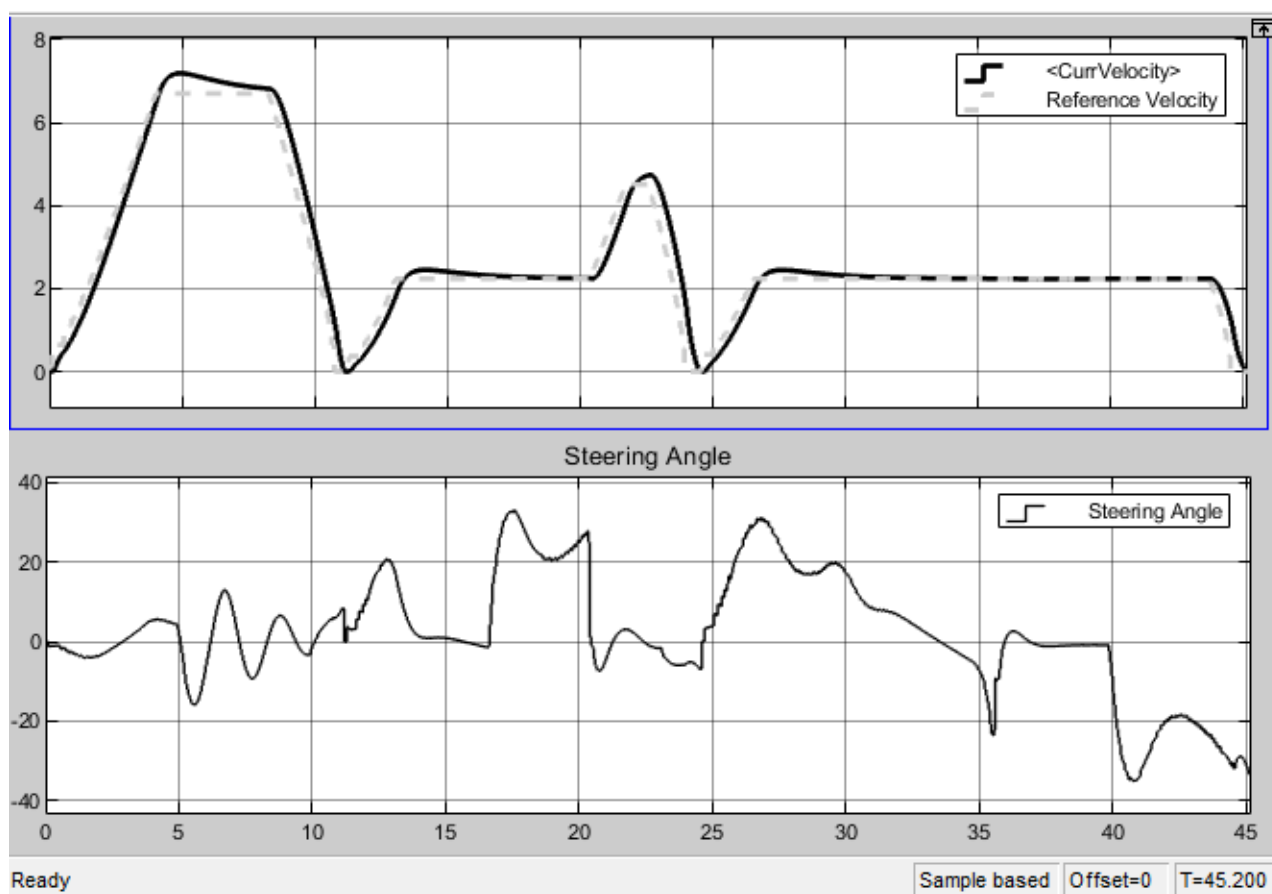


Рисунок 5.7 – Графік швидкості та кута повороту

5.3 Модель навколишнього середовища

Модель середовища представляє карту довкілля. Для системи паркування автомобілем ця карта включає наявні та зайняті місця для паркування, дорожню розмітку та перешкоди, такі як пішоходи чи інші транспортні засоби. Карти заповнення є загальним представленням для цієї форми моделі середовища. Така карта, як правило, будується з використанням одночасної локалізації та картографування (SLAM) шляхом інтеграції спостережень від датчиків лідара та камери. Розглянемо сценарій, коли карта вже надана, наприклад, системою транспортного засобу до інфраструктури (V2X) або камерами, що виходять на всі місця для паркування. Тут використовується статична карта стоянки і припускається, що само-локалізація автомобіля є точною.

Приклад паркінгу, використаний у цій роботі, складається з трьох шарів сітки:

- нерухомі перешкоди: цей шар містить нерухомі перешкоди, такі як стіни, перешкоди та межі стоянки;
- дорожня розмітка: цей шар містить інформацію про розмітку, включаючи розмітку на місцях для паркування;
- припарковані машини: цей шар містить інформацію про те, які місця для паркування вже зайняті.

Кожен шар карти містить різні види перешкод, які представляють різні рівні небезпеки для автомобіля, який рухається. Завдяки такій структурі кожен шар можна обробляти, оновлювати та підтримувати незалежно.

У кожному шарі темні клітини представляють зайняті клітини, а світлі клітини - вільні клітини. Для завантаження та відображення трьох шарів карти використовуємо функцію вказану нижче рисунок 5.8.

```
mapLayers = loadParkingLotMapLayers;
plotMapLayers(mapLayers)
```

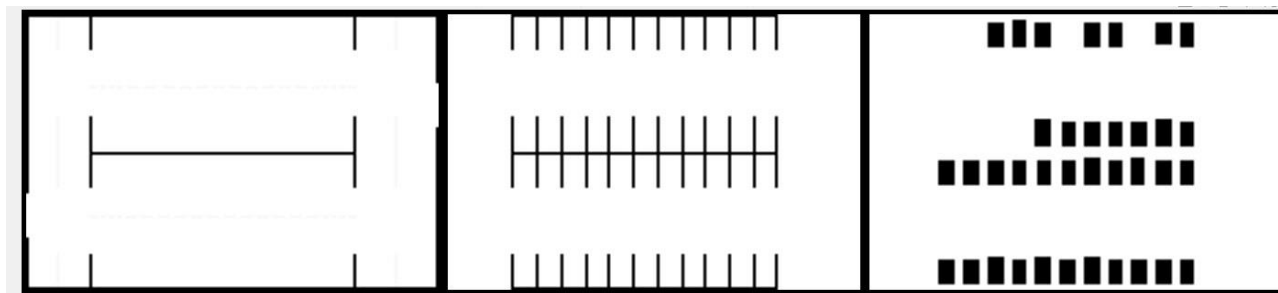



Рисунок 5.8 – Шари карти парковки

Для простоти та нормального сприйняття паркінгу об'єднуємо три шари в один.

```
costmap = combineMapLayers(mapLayers);
```

```
figure;
```

```
plot(costmap, 'Inflation', 'off');
```

```
legend off;
```

Результат представлений на рисунку 5.9

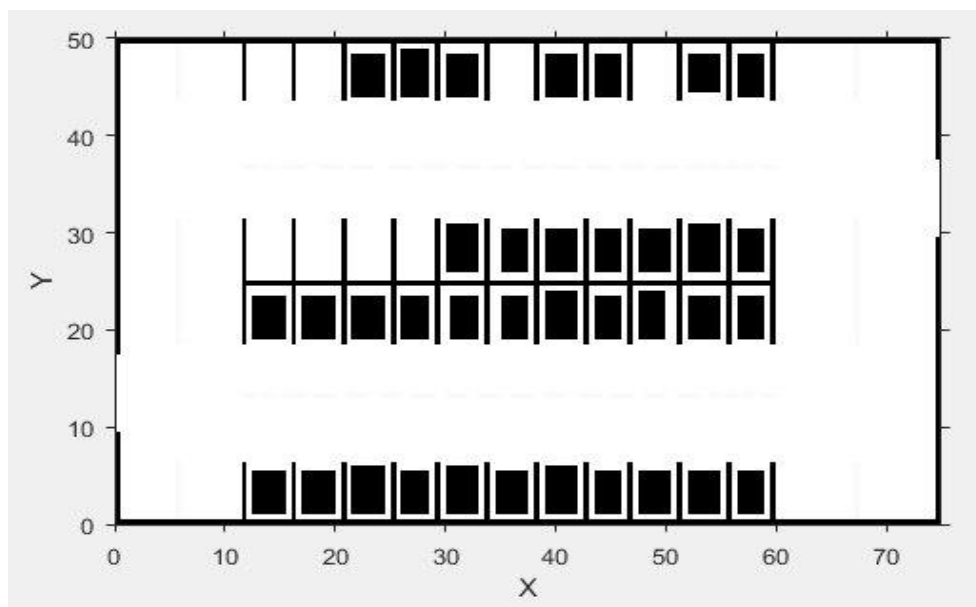


Рисунок 5.9 – Карта парковки

Комбінована карта – це карта парковки об'єкт, який представляє навколишнє середовище транспортного засобу як 2-D сітку заповнення. Кожна сітка в комірці має значення від 0 до 1, що відображає вартість навігації по комірці. Перешкоди мають вищу вартість, тоді як вільний простір – меншу. Клітина вважається

перешкодою, якщо її вартість вища за певне значення, і безкоштовною, якщо вартість нижча за певне значення.

Costmap охоплює всю площу стоянки, що розділена на квадратні осередки 0.5м-на-0.5м.

`costmap.MapExtent`; задаємо в метрах як ширина (x) та висота (y);

`costmap.CellSize`; розмір комірки також задаємо в метрах;

`ans = 0 75 0 50`;

`ans = 0,5000`;

Об'єкт `vehicleDimensions`, який створено для зберігання габаритів автомобіля, що буде автоматично паркуватися. Також потрібно визначити максимальний кут повороту автомобіля. Це значення визначає межі радіуса повороту під час планування та управління рухом.

`vehicleDims = vehicleDimensions`; транспортні розміри;

`maxSteeringAngle = 35`; максимальний кут повороту задаємо в градусах;

Оновлюємо габарити автомобіля – властивість перевірки зіткнення карти витрат з габаритами автомобіля для паркування. Цей параметр регулює ступінь заповнення на карті навколо перешкод відповідно до розміру припаркованого автомобіля, гарантуючи, що шлях без зіткнень буде знайдений.

`costmap.CollisionChecker.VehicleDimensions = vehicleDims`;

Стартову позицію транспортного засобу вказуємо як $[x, y, \theta]$, у світових координатах, (x, y) представляє положення центру задньої осі транспортного засобу у світовій системі координат, (θ) являє собою орієнтацію транспортного засобу щодо осі x .

`currentPose = [4 12 0]`;

5.4 Поведінковий рівень

Планування передбачає впорядкування всієї відповідної інформації в ієрархічні шари. Кожен наступний шар відповідає за більш дрібне завдання. Поведінковий шар знаходиться у верхній частині цього стеку. Він відповідає за

активацію та управління різними частинами місії, забезпечуючи послідовність навігаційних завдань. Поведінковий рівень збирає інформацію з усіх відповідних частин системи, включаючи:

- локалізація: поведінковий рівень перевіряє модуль локалізації для оцінки поточного місцезнаходження автомобіля;
- модель навколишнього середовища: системи сприйняття та синтезу датчиків подають карту довкілля навколо автомобіля;
- визначення місця для паркування: поведінковий рівень аналізує карту, щоб визначити найближче доступне місце для паркування;
- пошук глобального маршруту: модуль маршрутизації обчислює глобальний маршрут через дорожню мережу, отриманий або від служби картографування, або від інфраструктури V2X. Розкладання глобального маршруту як серії дорожніх зв'язків дозволяє по-різному планувати траєкторію руху кожного зв'язку. Наприклад, остаточний маневр паркування вимагає іншого профілю швидкості, ніж підхід до місця стоянки. У більш загальній обстановці це стає вирішальним для навігації вулицями, що мають різні обмеження швидкості, кількість смуг руху та дорожні знаки.

Замість того, щоб покладатися на датчики транспортних засобів для побудови карти навколишнього середовища, у даній роботі використовується карта, яка надходить із розумної автостоянки за допомогою зв'язку V2X. Для простоти припустимо, що карта має форму заповненої сітки з дорожніми посиланнями та місцями доступних місць для паркування, наданими V2X.

HelperBehavioralPlanner клас наслідує інтерфейс поведінкового рівня планування. HelperBehavioralPlanner створюється з допомогою карти і глобального плану маршруту. У роботі використовується статичний глобальний план маршруту, що зберігається в таблиці MATLAB, але зазвичай алгоритм маршрутизації, наданий локальною інфраструктурою паркування або службою картографування, визначає цей план. Глобальний план маршруту описується як послідовність відрізків смуги, яку потрібно проїхати до місця стоянки.

У таблиці 5.3 є три змінні: StartPose, EndPose, і Attributes. StartPosei; EndPose вказують початкову та кінцеву пози сегмента, виражені як $[x, y, \theta]$. Attributes визначає такі властивості сегмента, як обмеження швидкості.

```
data = load('routePlan.mat');
routePlan = data.routePlan;
routePlan = 4 × 3 table;
```

Таблиця 5.3 – План маршруту

StartPose	EndPose	Attributes
4,12,0	56,11,0	[1×1 struct]
56,11,0	70,19,90	[1×1 struct]
70,19,90	70,32,90	[1×1 struct]
70,32,90	53,39,180	[1×1 struct]

Прокладемо транспортний засіб у поточній позиції вздовж кожної цілі в плані маршруту.

Нижче наданий фрагмент коду що відображає поточну позицію авто та кожну наступну ціль відповідно до плану маршруту. Що зображено на рисунку 5.10

```
helperPlotVehicle(currentPose, vehicleDims, 'DisplayName', 'Current Pose');
legend;
for n = 1 : height(routePlan);
```

Кінцева позиція

```
vehiclePose = routePlan{n, 'EndPose'};
```

Розмітка позиції

```
legendEntry = sprintf('Goal %i', n);
helperPlotVehicle(vehiclePose, vehicleDims, 'DisplayName', legendEntry);
end;
hold off;
```

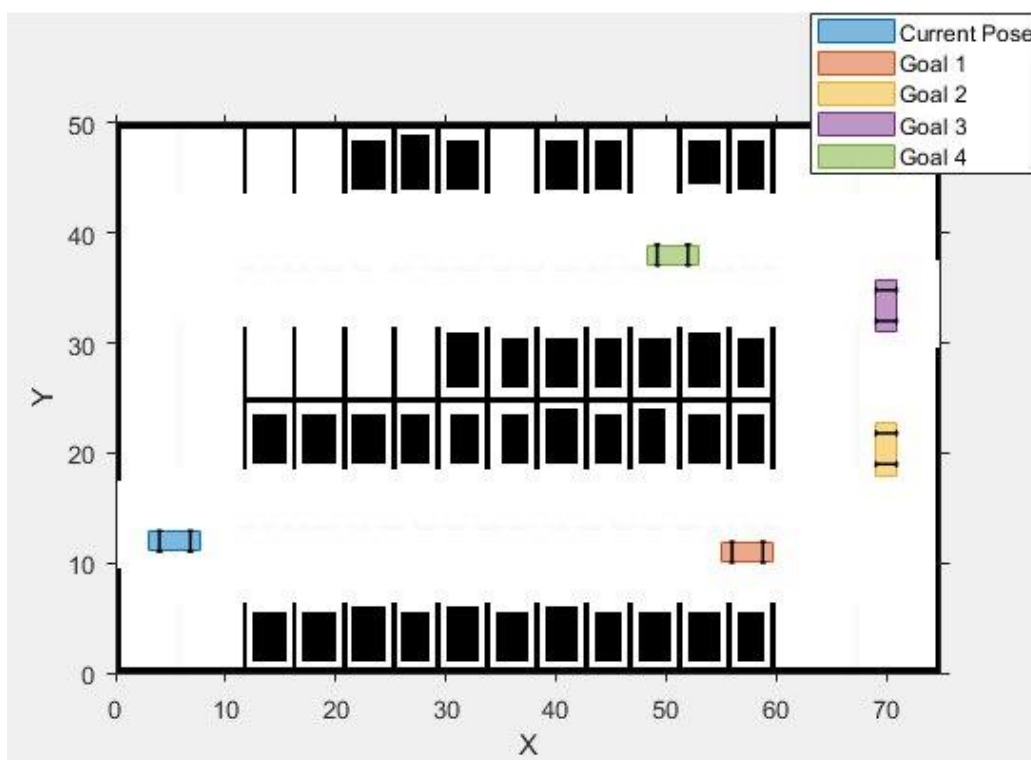


Рисунок 5.10 – Приклад розташування об’єктів на карті

Створимо допоміжний об’єкт планувальника поведінки. Метод `requestManeuver` запитує потік навігаційних завдань із планувальника поведінки до досягнення пункту призначення.

```
behavioralPlanner = HelperBehavioralPlanner (routePlan, maxSteeringAngle);
```

Транспортний засіб здійснює навігацію по кожному сегменту шляху, виконуючи такі дії:

- 1) планування руху: планування можливого шляху через карту, використовуючи оптимальний алгоритм швидкого вивчення випадкового дерева (RRT *) (`pathPlannerRRT`);
- 2) згладжування контуру: згладжуємо опорний шлях, пристосувавши до нього сплайни використовуючи `smoothPathSpline`;
- 3) генерація траєкторії: перетворюємо згладжений шлях у траєкторію, створивши профіль швидкості за допомогою `helperGenerateVelocityProfile`;
- 4) керування транспортним засобом: враховуючи згладжений контрольний шлях, `HelperPathAnalyzer` обчислює еталонну позицію та швидкість на основі поточної позиції та швидкості автомобіля. Наданий опорними значеннями,

`lateralControllerStanley` обчислює кут повороту для контролю напрямку руху автомобіля. `HelperLongitudinalController` обчислює команди прискорення та уповільнення для підтримання бажаної швидкості автомобіля;

5) перевірка цілей: перевіряємо, чи транспортний засіб досягнув кінцевої цілі за допомогою `helperGoalChecker`.

5.5 Планування руху

Враховуючи глобальний маршрут, планування руху можна використовувати для планування шляху через навколишнє середовище для досягнення кожної проміжної точки, поки транспортний засіб не досягне кінцевого пункту призначення. Запланований шлях для досягнення кожної цілі повинен бути здійсненим та без зіткнень. Доступним шляхом є той, який може бути реалізований транспортним засобом, враховуючи накладені на нього обмеження руху та динаміки. Система паркування автомобіля передбачає низькі швидкості та низькі прискорення. Це дозволяє безпечно ігнорувати динамічні обмеження, що виникають внаслідок інерційних ефектів.

Створення об'єкту `pathPlannerRRT` для конфігурації планувальника шляхів, використовуючи оптимальний підхід, що швидко вивчає випадкове дерево (RRT *). Сімейство алгоритмів планування RRT знаходить шлях за допомогою побудови дерева пов'язаних позицій транспортних засобів без зіткнень. Позиції з'єднуються за допомогою рульового управління Dubins або Reeds-Shepp, забезпечуючи, що сформований шлях є кінематично здійсненим.

```
motionPlanner = pathPlannerRRT(costmap, 'MinIterations', 1000,
    'ConnectionDistance', 10, 'MinTurningRadius', 20);
```

Планування шляху від поточної позиції до першої цілі за допомогою функції `plan`. Повернутий `driving.Path` об'єкт, `refPath` – це здійснений контрольний шлях до цілей без зіткнень.

```
goalPose = routePlan{1, 'EndPose'};
refPath = plan(motionPlanner, currentPose, goalPose);
```

Контрольний шлях складається з послідовності сегментів шляху. Кожен сегмент шляху описує набір маневрів Dubins або Reeds-Shepp, використовуваних для підключення до наступного сегмента. Оглянемо масив сегмента шляху Dubins з властивостями.

```
refPath.PathSegments;
```

```
ans =
```

```
1×6 DubinsPathSegment array with properties:
```

```
StartPose;
```

```
GoalPose;
```

```
MinTurningRadius;
```

```
MotionLengths;
```

```
MotionTypes;
```

```
Length;
```

Контрольний шлях містить позиції переходу на шляху, що представляють точки вздовж шляху, що відповідають переходу від одного маневру до наступного. Вони також можуть представляти зміни напрямку, наприклад, від прямого до зворотного руху по шляху Reeds-Shepp.

Отримаємо перехідні позиції та команди із запланованого шляху.

```
[transitionPoses, directions] = interpolate(refPath);
```

Візуалізуємо запланований шлях за допомогою функції plot. Результат планування маршруту зображено на рисунку 5.11.

```
plot(motionPlanner);
```

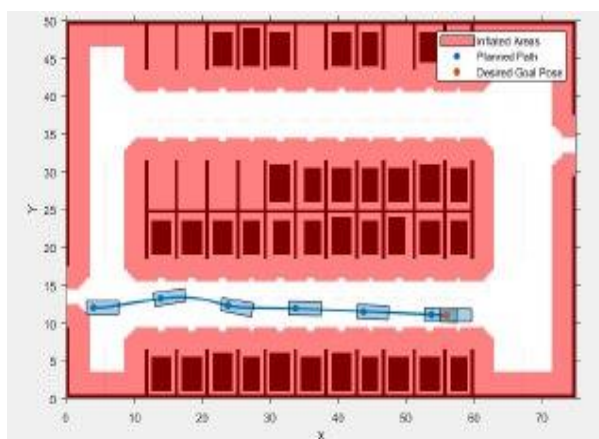


Рисунок 5.11 – Планування маршруту на карті заповнення

Червоні ділянки представляють ділянки карти витрат, з якими початок транспортного засобу (центр задньої осі) не повинен перетинатися, щоб уникнути наїзду на будь-які перешкоди. pathPlannerRRT знаходить шляхи, які дозволяють уникнути перешкод, перевіряючи, щоб створені позиції транспортного засобу не лежали на цих ділянках.

5.6 Згладжування контуру та генерація траєкторії

Контрольний шлях, згенерований планувальником шляхів, складається або з сегментів Dubins або Reeds-Shepp. Кривизна на місцях з'єднання двох таких сегментів не є суцільною і може призвести до різких змін кута повороту. Щоб уникнути такого неприродного руху шлях повинен бути безперервно диференційованим і, отже, рівним. Один із підходів до згладжування контуру передбачає підгонку параметричного кубічного сплайну. Сплайн-фітинг дозволяє генерувати плавний шлях, який може виконувати контролер.

Використовуємо smoothPathSpline для розміщення параметричного кубічного сплайну, який проходить через усі точки переходу контрольного шляху. Сплайн приблизно відповідає початковому і кінцевому напрямкам з початковим і кінцевим кутом руху автомобіля.

Для визначення кількості позицій для повернення, використовуємо відстань приблизно 0.1 м.

```
approxSeparation = 0.1;
```

```
numSmoothPoses = round(refPath.Length / approxSeparation);
```

Повернення дискретизованих позицій по згладженому шляху

```
[refPoses, directions, cumLengths, curvatures] = smoothPathSpline(transitionPoses,
directions, numSmoothPoses);
```

Для відображення згладженого шляху використовуємо функцію наведену нижче, результат згладжування на рисунку 5.12.

```
hSmoothPath = plot(refPoses(:, 1), refPoses(:, 2), 'r', 'LineWidth', 2, ...
'DisplayName', 'Smoothed Path');
```

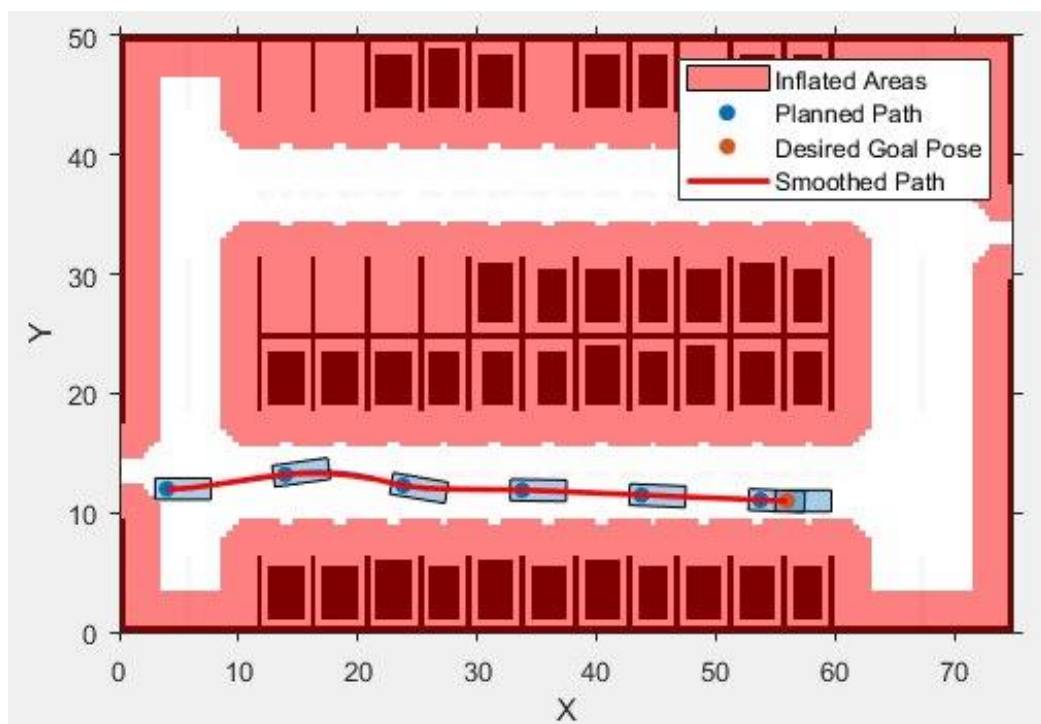



Рисунок 5.12 – Згладжування траєкторії руху

Далі перетворимо згенерований плавний шлях у траєкторію, яку можна виконати за допомогою профілю швидкості. Розрахуємо профіль швидкості для кожного шляху у вигляді послідовності з трьох фаз: розгін до заданої максимальної швидкості, підтримка максимальної швидкості та уповільнення до кінцевої швидкості. Функція `helperGenerateVelocityProfile` генерує такий профіль швидкості.

Вкажемо початкову, максимальну та кінцеву швидкості, щоб транспортний засіб набирає швидкість для руху з 0 метрів/секунду, розганявся до швидкості 5 метрів / секунду і зупинявся скидаючи швидкість до 0 метрів/секунду .

```
maxSpeed = 5;
```

```
startSpeed = 0;
```

```
endSpeed = 0;
```

Далі формуємо профіль швидкості

```
refVelocities = helperGenerateVelocityProfile(directions, cumLengths, curvatures,  
startSpeed, endSpeed, maxSpeed);
```

refVelocities містить еталонні швидкості для кожної точки вздовж згладженого шляху. Побудуємо графік сформованого профілю швидкості, який зображено на рисунку 5.13.

```
plotVelocityProfile (cumLengths, refVelocities, maxSpeed);
```

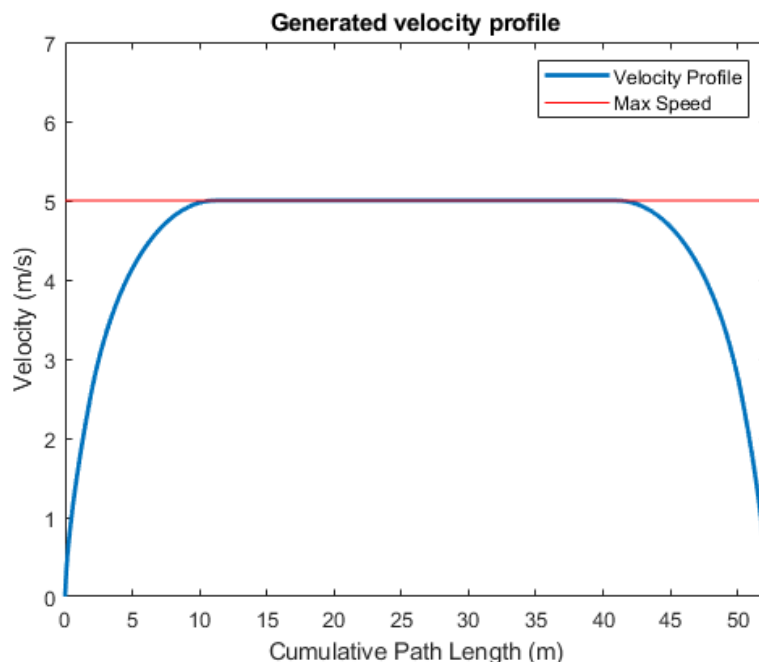


Рисунок 5.13 – Графік профілю швидкості

5.7 Виконання повного плану

Тепер об'єднаємо усі попередні кроки в процесі планування та запустимо моделювання для повного плану маршруту. Цей процес передбачає включення планувальника поведінки.

Повернення транспортного засобу до початкової вихідної точки що описувалась вище (x , y , θ)

```
currentPose = [4 12 0];
```

```
vehicleSim.setVehiclePose(currentPose);
```

Скидання швидкості після досягнення пункту призначення тобто кінцевої цілі перед виконанням маневру паркування.

```
currentVel = 0; m\c
```

```
vehicleSim.setVehicleVelocity(currentVel);
```

```
while ~reachedDestination(behavioralPlanner);
```

Запит наступного маневру у поведінкового рівня

```
[nextGoal, plannerConfig, speedConfig] = requestManeuver(behavioralPlanner, ...
    currentPose, currentVel);
```

Налаштування планувальника руху

```
configurePlanner(motionPlanner, plannerConfig);
```

Планування опорного шляху за допомогою планувальника до наступної цілі

```
refPath = plan(motionPlanner, currentPose, nextGoal);
```

Перевірка правильності шляху, якщо планувальнику не вдається обчислити шлях або шлях не є без зіткнень через оновлення карти.

```
isReplanNeeded = ~checkPathValidity(refPath, costmap);
```

```
if isReplanNeeded;
```

```
    warning('Unable to find a valid path. Attempting to re-plan.');
```

Перепланування поведінки

```
replanNeeded(behavioralPlanner);
```

```
    continue;
```

```
end;
```

Отримання перехідних позицій та команд із запланованого шляху

```
[transitionPoses, directions] = interpolate(refPath);
```

Згладити шлях

```
numSmoothPoses = round(refPath.Length / approxSeparation);
```

```
[refPoses, directions, cumLengths, curvatures] =
```

```
smoothPathSpline(transitionPoses, directions, numSmoothPoses);
```

Формування профілю швидкості

```
refVelocities = helperGenerateVelocityProfile(directions, cumLengths,
    curvatures, startSpeed, endSpeed, maxSpeed);
```

Налаштування аналізатора шляху

```
pathAnalyzer.RefPoses = refPoses;
```

```
pathAnalyzer.Directions = directions;
```

```
pathAnalyzer.VelocityProfile = refVelocities;
```

Скидання поздовжнього контролера

```
reset(lonController);
```

```
reachGoal = false;
```

Виконання циклу управління

```
while ~reachGoal;
```

Знаходження опорної позиції на шляху та відповідної швидкості

```
[refPose, refVel, direction] = pathAnalyzer(currentPose, currentVel);
```

Оновлення напрямку руху моделі авто

```
updateDrivingDirection(vehicleSim, direction);
```

Обчислення команди керування

```
steeringAngle = lateralControllerStanley(refPose, currentPose, currentVel, ...  
'Direction', direction, 'Wheelbase', vehicleDims.Wheelbase);
```

Обчислення команди прискорення та уповільнення

```
lonController.Direction = direction;
```

```
[accelCmd, decelCmd] = lonController(refVel, currentVel);
```

Моделювання транспортного засобу за допомогою виходів контролера

```
drive(vehicleSim, accelCmd, decelCmd, steeringAngle);
```

Перевірка чи транспортний засіб досягає мети

```
reachGoal = helperGoalChecker(nextGoal, currentPose, currentVel,  
speedConfig.EndSpeed, direction);
```

Отримання поточної позиції та швидкості руху автомобіля

```
currentPose = getVehiclePose(vehicleSim);
```

```
currentVel = getVehicleVelocity(vehicleSim);
```

```
end;
```

```
end;
```

Моделювання руху авто представлено на рисунку 5.14.

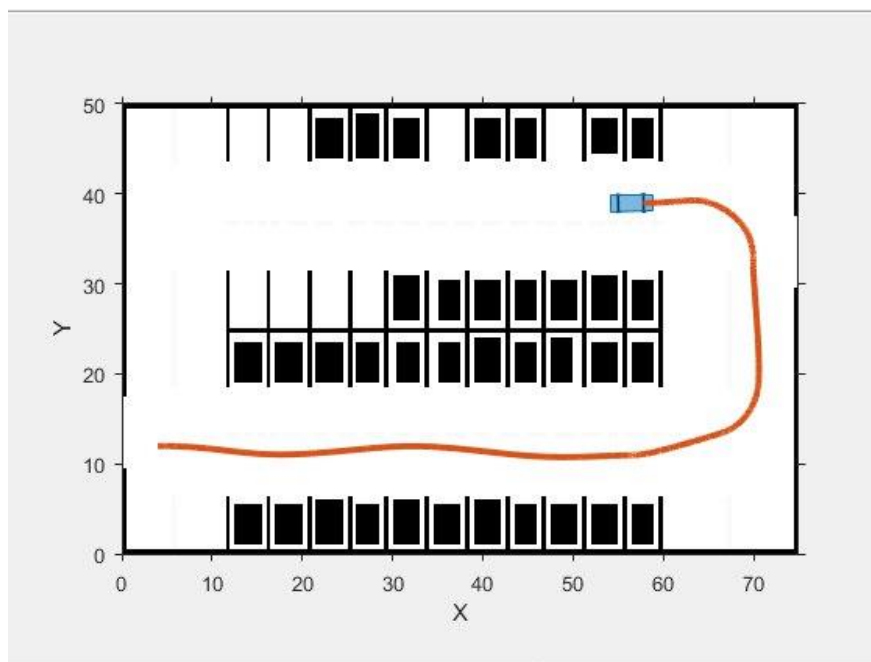


Рисунок 5.14 – Моделювання руху авто

5.8 Маневр паркування

Тепер, коли транспортний засіб знаходиться біля місця для паркування, для його паркування використовується спеціальний маневр для паркування. Цей маневр вимагає проходження вузьким коридором, обмеженим краями місця для паркування з обох кінців. Такий маневр, як правило, супроводжується ультразвуковими датчиками або лазерними сканерами, які постійно перевіряють наявність перешкод.

`vehicleCostmap` використовує перевірку зіткнень на основі порівняння значень комірки. Спочатку оглянемо поточну перевірку зіткнень, яка використовується, що зображена на рисунку 5.15.

```
ccConfig = costmap.CollisionChecker;
figure;
plot(ccConfig);
title('Current Collision Checker');
```

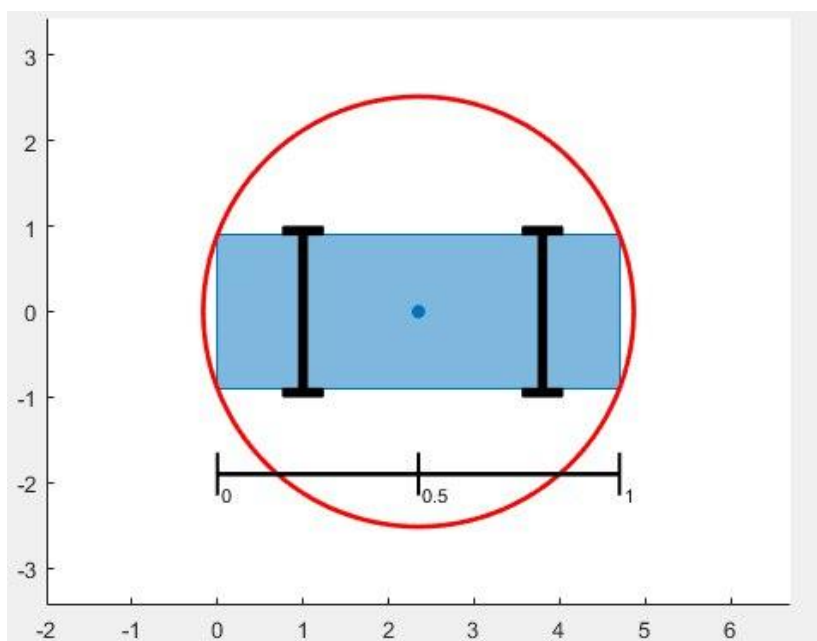


Рисунок 5.15 – Перевірка можливості зіткнення

Перевірка зіткнень виконується шляхом заповнення перешкод на карті витрат радіус заповнення та перевірки, залежить від радіуса кола, показаного вище на рисунку 5.15. на завищеній комірці сітки. Остаточний маневр паркування вимагає більш точного, менш консервативного механізму перевірки зіткнень. Це зазвичай вирішується шляхом представлення форми транспортного засобу за допомогою кількох (3-5) кіл, що перекриваються, замість одного кола.

Використаємо більшу кількість кіл у засобі перевірки зіткнень та візуально оглянемо на рисунку 5.16 перевірку зіткнень. Це дозволяє планувати вузькі проходи.

```
ccConfig.NumCircles = 4;
figure;
plot(ccConfig);
title('New Collision Checker');
```

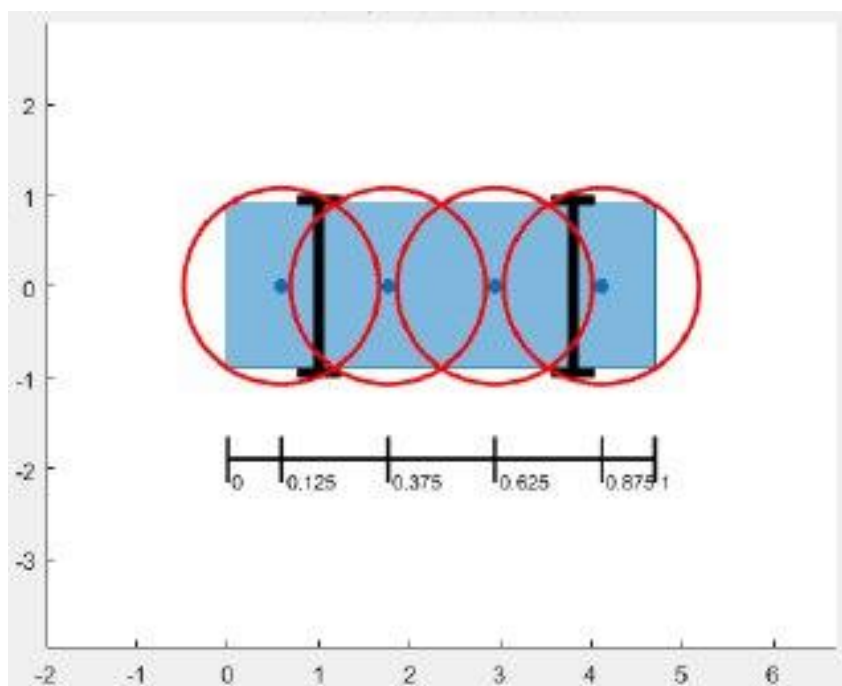


Рисунок 5.16 – Перевірка можливості зіткнень

Радіус надування зменшився, що дозволило планувальнику знайти безперешкодний шлях до місця стоянки, це можна побачити на рисунку 5.17.

```
figure;
```

```
plot(costmap);
```

```
title('Costmap with updated collision checker');
```

Визначення бажаної позиції в місці для паркування повернутою системою V2X

```
parkPose = [36 44 90];
```

```
preParkPose = currentPose;
```

Обчислення необхідного маневру для паркування

```
refPath = plan(parkMotionPlanner, preParkPose, parkPose);
```

Відображення отриманого маневру паркування зображено на рисунку 5.18.

```
figure;
```

```
plotParkingManeuver(costmap, refPath, preParkPose, parkPose);
```

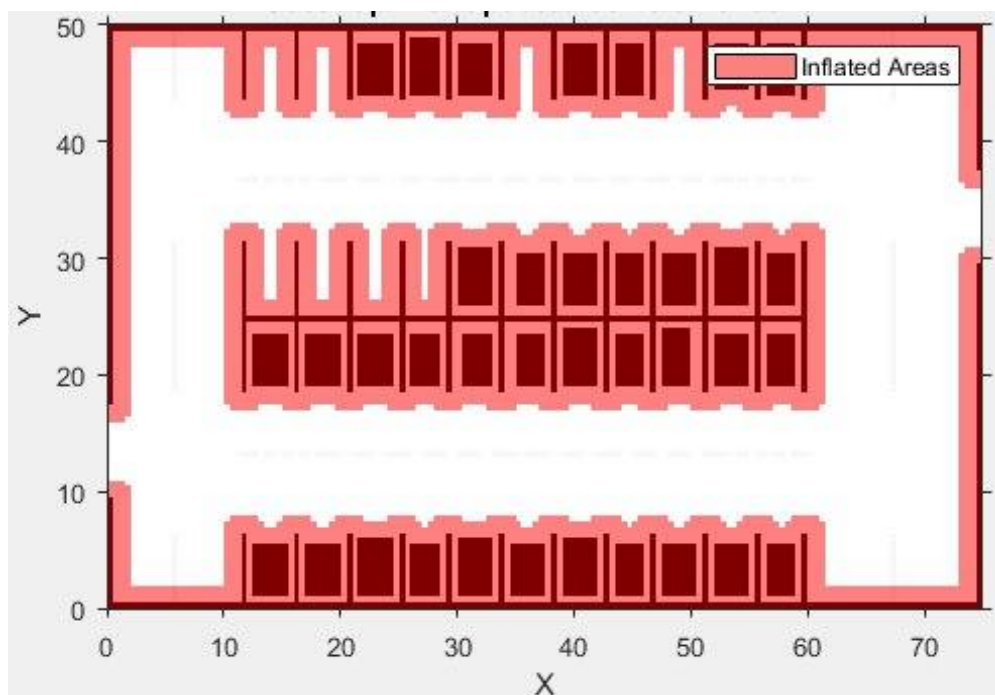


Рисунок 5.17 – Оновлена карта витрат (заповнення перешкод)

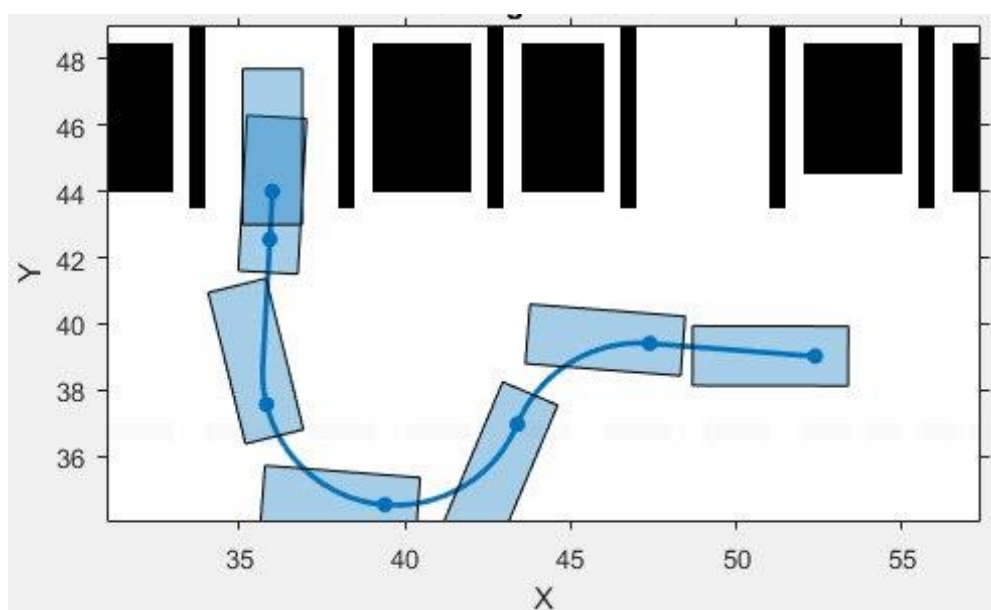


Рисунок 5.18 – Маневр паркування

Як тільки маневр буде знайдений, повторюємо попередній процес, щоб визначити повний план: згладжуємо траєкторію, формуємо профіль швидкості та дотримуємось траєкторії за допомогою контролера зворотного зв'язку.

Отримати перехідні позиції та команди з запланованого шляху

```
[transitionPoses, directions] = interpolate(refPath);
```


Згладити шлях

```
numSmoothPoses = round(refPath.Length / approxSeparation);
[refPoses, directions, cumLengths, curvatures] = smoothPathSpline(transitionPoses,
directions, numSmoothPoses);
```

Налаштування профілю генератора швидкості для зупинки в кінцевій траєкторії, з обмеженням швидкості 5 миль/год

```
refVelocities = helperGenerateVelocityProfile(directions, cumLengths, curvatures,
currentVel, 0, 2.2352);
pathAnalyzer.RefPoses = refPoses;
pathAnalyzer.Directions = directions;
pathAnalyzer.VelocityProfile = refVelocities;
```

Скидання поздовжнього контролера

```
reset(lonController);
reachGoal = false;
while ~reachGoal;
```

Знаходження опорної пози на шляху та відповідної швидкості

```
[refPose, refVel, direction] = pathAnalyzer(currentPose, currentVel);
```

Оновлення напрямку руху моделі авто

```
updateDrivingDirection(vehicleSim, direction);
```

Обчислення команд керування

```
steeringAngle = lateralControllerStanley(refPose, currentPose, currentVel, ...
'Direction', direction, 'Wheelbase', vehicleDims.Wheelbase);
```

Обчислення команд прискорення та уповільнення

```
lonController.Direction = direction;
[accelCmd, decelCmd] = lonController(refVel, currentVel);
```

Імітація транспортного засобу за допомогою виходів контролера

```
drive(vehicleSim, accelCmd, decelCmd, steeringAngle);
```

Перевірка чи досягає транспортний засіб мети

```
reachGoal = helperGoalChecker(parkPose, currentPose, currentVel, 0, direction);
```

Отримання поточної позиції та швидкості руху автомобіля

```

currentPose = getVehiclePose(vehicleSim);
currentVel = getVehicleVelocity(vehicleSim);
end;

```

Вікно моделювання паркування автомобіля, зображено на рисунку 5.19.

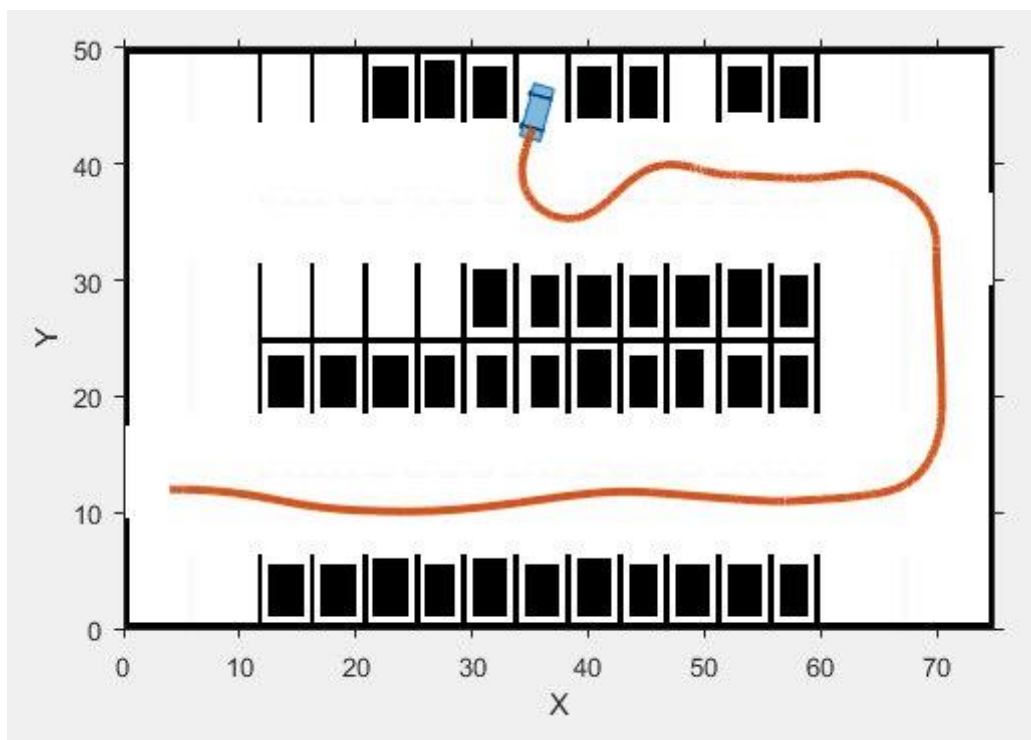


Рисунок 5.19 – Вікно моделювання паркування

Альтернативний спосіб припаркувати транспортний засіб – повернутися на місце кінцевої цілі перед маневром паркування, планувальник руху повинен використовувати метод підключення Reeds-Shepp для пошуку можливого шляху. Метод підключення Reeds-Shepp дозволяє здійснювати зворотні рухи під час планування. Що не потребує взяття великого радіусу для маневру паркування, та виїзду на полосу зустрічного руху коли це не можливо.

Вказання позиції паркування, яка відповідає маневрові паркування

```
parkPose = [49 47 -90];
```

Зміна методу підключення, щоб дозволити зворотні рухи

```
parkMotionPlanner.ConnectionMethod = 'Reeds-Shepp';
```

Щоб знайти можливий шлях, планувач руху потрібно відрегулювати. Використовуємо більший радіус повороту та відстань з'єднання, щоб забезпечити плавний рух задом.

```
parkMotionPlanner.MinTurningRadius = 10;
```

```
parkMotionPlanner.ConnectionDistance = 15;
```

Скидання позиції та швидкості автомобіля

```
currentVel = 0;
```

```
vehicleSim.setVehiclePose(preParkPose);
```

```
vehicleSim.setVehicleVelocity(currentVel);
```

Обчислення маневру паркування

```
replan = true;
```

```
while replan;
```

```
refPath = plan(parkMotionPlanner, preParkPose, parkPose);
```

Шлях, відповідний маневру паркування, невеликий і вимагає точного маневрування. Замість інтерполяції лише в перехідних позах точніше інтерполювати по довжині шляху.

```
numSamples = 10;
```

```
stepSize = refPath.Length / numSamples;
```

```
lengths = 0 : stepSize : refPath.Length;
```

```
[transitionPoses, directions] = interpolate(refPath, lengths);
```

Перепланування, якщо шлях містить більше ніж одну позицію перемикавання напрямків або якщо шлях занадто довгий

```
replan = sum(abs(diff(directions)))~>2 || refPath.Length > 20;
```

```
end;
```

Маневр паркування представлений на рисунках 5.20, 5.21.

```
plotParkingManeuver(costmap, refPath, preParkPose, parkPose);
```

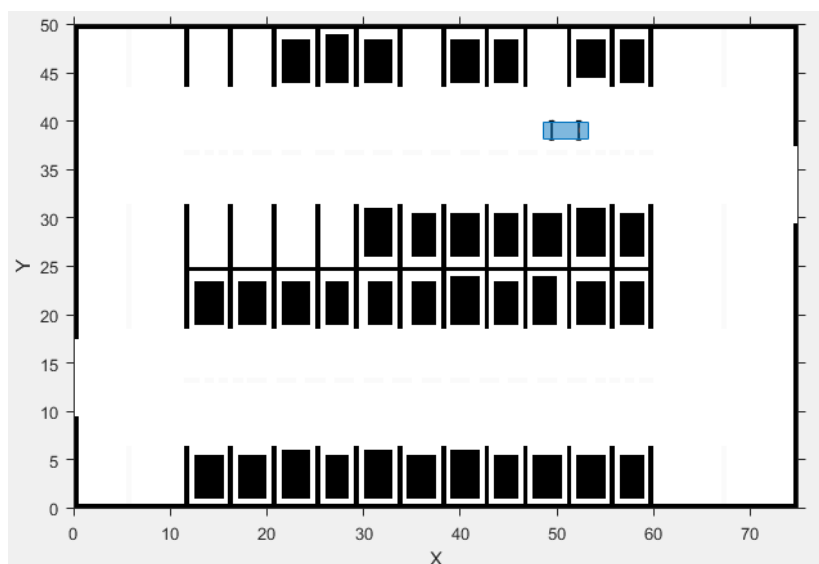


Рисунок 5.20 – Маневр паркування заднім ходом

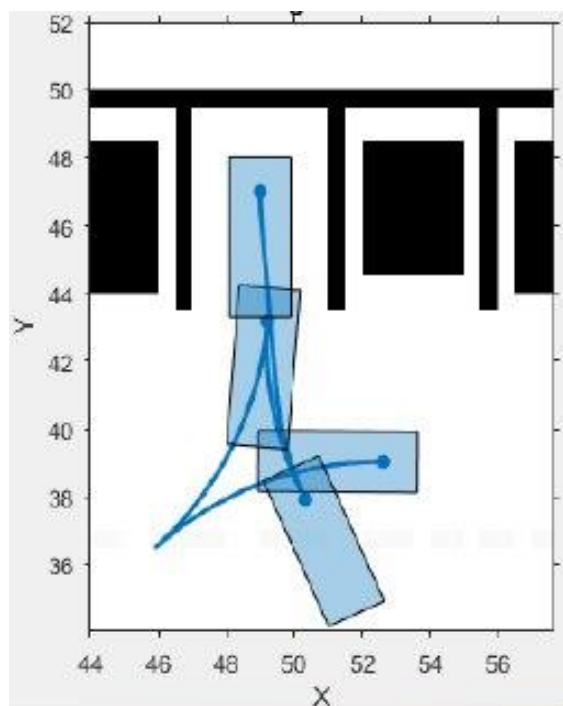


Рисунок 5.21 – Паркування заднім ходом

Як тільки маневр буде знайдений згладжуємо траєкторію, формуємо профіль швидкості та дотримуємось траєкторії за допомогою контролера зворотного зв'язку. Кінцевий рух моделювання паркування заднім ходом відображений на рисунку 5.22.

Згладжуємо траєкторію

```
numSmoothPoses = round(refPath.Length / approxSeparation);
```

```
[refPoses, directions, cumLengths, curvatures] = smoothPathSpline(transitionPoses,
directions, numSmoothPoses, 0.5);
```

Формування профілю швидкості

```
refVelocities = helperGenerateVelocityProfile(directions, cumLengths, curvatures,
currentVel, 0, 2);
```

```
pathAnalyzer.RefPoses = refPoses;
```

```
pathAnalyzer.Directions = directions;
```

```
pathAnalyzer.VelocityProfile = refVelocities;
```

Отримання поточного напрямку руху

```
currentDir = getDrivingDirection(vehicleSim);
```

Знаходження опорної позиції на шляху та відповідну швидкість.

```
[refPose, refVel, direction] = pathAnalyzer(currentPose, currentVel);
```

Якщо транспортний засіб змінює напрямок руху, відбувається скидання швидкості авто в моделі і скидання поздовжнього контролера

```
if currentDir ~= direction;
```

```
    currentVel = 0;
```

```
    setVehicleVelocity(vehicleSim, currentVel);
```

```
    reset(lonController);
```

```
end;
```

Оновлення напрямку руху моделі якщо автомобіль змінює напрям руху, скидання і повернення поточної швидкості авто як нуль.

```
currentVel = updateDrivingDirection(vehicleSim, direction, currentDir);
```

Обчислення команди керування

```
steeringAngle = lateralControllerStanley(refPose, currentPose, currentVel, ...
```

```
    'Direction', direction, 'Wheelbase', vehicleDims.Wheelbase);
```

Обчислення команди прискорення та уповільнення

```
lonController.Direction = direction;
```

```
[accelCmd, decelCmd] = lonController(refVel, currentVel);
```

Моделювання транспортного засобу за допомогою виходів контролера

```
drive(vehicleSim, accelCmd, decelCmd, steeringAngle);
```

Перевірка чи досягає транспортний засіб мети

```
reachGoal = helperGoalChecker(parkPose, currentPose, currentVel, 0, direction);
```

Отримання поточної позиції та швидкості автомобіля

```
currentPose = getVehiclePose(vehicleSim);
```

```
currentVel = getVehicleVelocity(vehicleSim);
```

```
end;
```

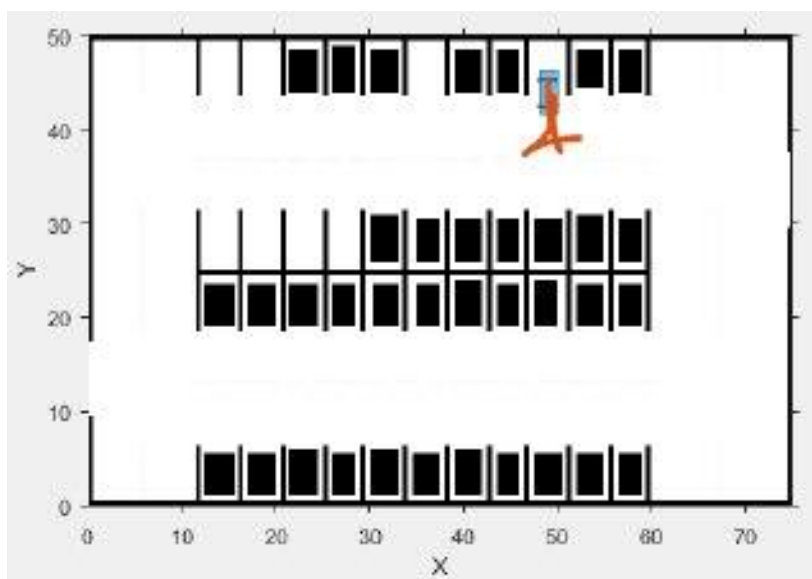


Рисунок 5.22 – Моделювання паркування заднім ходом

5.9 Висновки до розділу

Отримані наступні результати:

- 1) планування можливого шляху у напівструктурованому середовищі, наприклад, на автостоянці, використовуючи алгоритм планування шляху RRT;
- 2) згладжування контуру за допомогою сплайнів і формування профілю швидкості вздовж згладженого контуру;
- 3) керування транспортним засобом, щоб пройти контрольний шлях із бажаною швидкістю;
- 4) реалізовано різні способи паркування, використовуючи різні налаштування планувальника руху.

6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

6.1 Опис ідеї проекту

Таблиця 6.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення системи автоматичного паркування автомобілів на базі технології інтернету речей	1. Управління транспортними засобами на територіях паркінгів	1. Облік ТЗ 2. Контроль пересування ТЗ 3. Планування короткого маршруту, маневру паркування 4. Запобігання зіткнення ТЗ 5. Попередження про зіткнення зміна маршруту або ж негайна зупинка ТЗ 6. Створення трьох шарової мапи паркінгу.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1 FidPark-CAR	Конкурент2 AFAPARK			
1.	Технологія	Bluetooth Для точного	Інформаційні табло для	Інформаційні табло для			Найбільше підход

№ п/ п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів			W (слаб ка сторо на)	N (нейтрал ьна сторона)	S (сильна сторон а)
		Мій проект	Конкурент1 FidPark- CAR	Конкурент2 AFAPARK			
		позиціонування безпілотного авто на території паркінгу	допомоги водієві з навігацією по паркінгу	допомоги водієві з навігацією по паркінгу			ить для даної галузі
2.	Зона охоплення	900 м	Поки знаходиться у полі зору водія	Поки знаходиться у полі зору водія			Більша зона покриття
3.	Точність	до 1 м	Зона відповідальності водія авто	Зона відповідальності водія авто		Достатня для даної галузі застосування	
4.	Швидкість передачі даних	Середня Залежить від завантаженості паркінгу	Середня Залежить від завантаженості паркінгу	Середня Залежить від завантаженості паркінгу		Як у основног о конкурента	
5.	Завадостійкість	Висока	Відсутня у систем такого типу	Відсутня у систем такого типу		Як у конкурентів	

№ п/п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів			W (слаб ка сторо на)	N (нейтрал ьна сторона)	S (сильна сторон а)
		Мій проект	Конкурент1 FidPark- CAR	Конкурент2 AFAPARK			
6.	Рівень споживанн я енергії	Низький	Низький	Низький		Як у конкурентів	

6.2 Технологічний аудит ідеї проекту

Таблиця 6.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.		GSM	+	-
2.		Wi-Fi	+	+
3.		BLE	+	+
4.		NanoLOC	+	-
5.		ZigBee	+	-
6.		UWB	+	-
Обрана технологія реалізації ідеї проекту: Побудова радіомережі для визначення позиції за допомогою приймачів Bluetooth Low Energy (LE). Ця технологія є доступною з точки зору апаратної і програмної реалізації і задовольняє вимогам з точності, завадостійкості				

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	2000000000 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає обмежень
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	70%

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Системи контролю та управління ТЗ територіях паркінгів	Представники будівельної, логістичної, сфери	Основна ціль всіх груп клієнтів однакова – зручність, економія часу, контроль місцезнаходження ТЗ	Система повинна бути автоматизованою, простою у використанні, надійною

Таблиця 6.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Поява нових конкурентів на ринку	Зменшення долі на ринку	Збільшення витрат на рекламу, створення нових технологічних рішень
2.	Знецінення гривні	Зменшення попиту на ринку України	Вливання на іноземний ринок

Таблиця 6.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Попит на системи позиціонування в автомобільній сфері, або в сферах з аналогічними потребами	Збільшення долі ринку, можливість розвитку нових систем	Збільшення обсягів виробництва, впровадження нових технологій
2.	Зменшення ціни на електричні компоненти та складові системи	Зменшення собівартості продукту, що призведе до зростання попиту на більш дешевий продукт	Вихід на нові ринки збуту, збільшення обсягів продаж

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції – чиста	На ринку присутні конкурентні компанії зі схожими технологіями	Боротьба за першість що призводить до зменшення собівартості, вдосконалення технологій, пошуку нових ідей розвитку
2. За рівнем конкурентної боротьби – глобальний	Вихід на зовнішні ринки, відсутність локальних конкурентів	Продаж продукції за мінімально можливою ціною
3. За галузевою ознакою – внутрішньогалузева	Конкуренція на ринку ведеться в будівельній, логістичній галузях	Необхідно зосередити зусилля на пошуку конкурентних переваг, які дозволять компанії займати стійкі конкурентні позиції на даному ринку
4. Конкуренція за видами товарів – товарно-родова	Конкуренція на рівні технології задоволення потреб. Існує конкуренція з іншими компаніями	Ведення конкурентної боротьби в технологічній сфері
5. За характером конкурентних переваг – нецінова	Для значної частки споживачів визначальною при виборі є технічна перевага, або забезпечення	Ведення конкурентної боротьби в технологічній сфері

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
	технічних вимог	
6. За інтенсивністю – марочна		Диференціація систем

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	AFAPARK FIDPARK	Відсутні бар'єри входження в ринок	Цінові Значення розміру поставок – відсутні\цінові	Фактор вибору	Розширення функціоналу продукту конкурента
Висновки:	Інтенсивність конкурентоспроможності – висока	- є можливості входу в ринок; - наявно 2 головних конкуренти; - строки виходу на ринок: середньо-строкові	Постачальники не диктують умов	Споживач обирає необхідний йому технічні характеристик и, можливість видозмінення системи, Приваблива ціна	Обмеження для роботи на ринку через товари замінники

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Ціновий	Ціна на продукт
2.	Продуктова диференціація	Наявність різновиду пристрою за функціоналом
3.	Динаміка галузі	Наявність попиту на продукт
4.	Продуктова лінія	Наявність видових різновидів пристрою

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін «проекту»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з продуктом						
			-3	-2	-1	0	+1	+2	+3
1.	Ціновий	5							
2.	Продуктова диференціація	10							
3.	Динаміка галузі	15							
4.	Продуктова лінія	10							

Таблиця 6.12 – SWOT-аналіз стартап-проекту

Сильні сторони: Нові технології Надійне обладнання Технічні характеристики продукту Функціональність Ресурси	Слабкі сторони: Відсутність досліджень з приводу аналізу нових каналів збуту
Можливості: Зростання ринку Підтримка зі сторони інвесторів Зарубіжний ринок	Загрози: Внутрішній ринок

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Використання засобів стимулювання збуту та мерчандайзингу для збільшення продаж, представлення продукту на різних конференціях	Отримання ресурсів від інвестицій	До 2-х років
2.	Розширення асортименту за рахунок створення нових продуктів або вдосконалення існуючого	Отримання ресурсів від інвестицій	До 3-х років
3.	Збільшення представленості на міжнародній арені	Отримання довгострокових кредитів	До 5 років

6.4 Розроблення ринкової стратегії проекту

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
	Концентрований маркетинг	Вузькоспеціалізовані потреби	Орієнтований попит – великий	Конкуренція представлена 2-ма виробниками регіонального значення	Середня з майбутньою перспективою
Які цільові групи обрано: було обрано концентрований маркетинг – робота із спеціалізованими галузями, для яких потрібно вирішувати задачі паркування					

Таблиця 6.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Конкурентна стратегія	Посилити використання таких конкурентних переваг як унікальне позиціонування, паркування та рівня продуктової диференціації	Посилення тиску зі сторони товарів конкурентів	Стратегія спеціаліза ції
2.	Цінова стратегія	Даний метод дозволить регулювати ціни в залежності від витрат виробництва, а також забезпечити цільовий прибуток	1.Регулювання цін на логістичну діяльність з боку держави; 2.Дефіцит якісного людського персоналу; 3.Зростання цін на витратні матеріали; 4.Коливання попиту.	Стратегія лідерства по витратах
3.	Стратегія просування	Для збільшення обсягів реалізації, формування лояльності споживачів	1. Зростання числа клієнтів; 2. Залежність попиту від купівельної спроможності споживачів 3.Збільшення інтенсивності конкуренції між існуючими гравцями на ринку	Стратегія спеціаліза ції

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
	Ні	Компанія буде шукати нових споживачів	Компанія не буде копіювати основні характеристики товару	Стратегія заняття конкурентної ніші

Таблиця 6.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
	1. Точність 2. Зона охопту 3. Масштабування 4. Прийнятна ціна	Стратегія спеціаліз ації	1. Велика зона охопту 2. Точність 3. Завадостійкість 4. Доступність технології	1. Ціна 2. Надійність 3. Контроль

6.5 Розроблення маркетингової програми стартап-проекту

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1.	Ціна	Дешевше, ніж у конкурентів	Дешевше, ніж у конкурентів
2.	Якість	Точність позиціонування, завадостійкість, надійність	Точність відповідає технічним вимогам галузі, технології передачі даних є стійкими до спотворення. Часткове дублювання системи на випадок відмови головних елементів
3.	Дальність	Більше покриття	Більша зона охопту базової станції призводить до зменшення кількості базових станцій на відміну від конкурентів

Таблиця 6.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система автоматичного паркування автомобілів на основі систем локального позиціонування		
	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Базові станції	Нм	Тх
	2. Мітки	Нм	Тх
	3. Живлення (електроенергія)	М	Тл
	4. Автоматизована система	Нм	Тл
	Якість: тестування за державними і міжнародними стандартами		
	Пакування: без пакування, розгортання та налаштування		

Рівні товару	Сутність та складові
	системи на об'єкті
	Марка: в процесі розробки
	До продажу: інформацію про товар можна дізнатися у менеджерів та технічних консультантів
	Після продажу: повністю готовий для роботи
За рахунок чого потенційний товар буде захищено від копіювання: товар буде захищено за рахунок інтелектуальної власності і новітніх розроблених систем (ноу-хау)	

Таблиця 6.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	Більше 1000000 грн	Більше 1000000 грн	Від 10 млн. грн до 50 млн. грн	Від 500000 до 1000000

Таблиця 6.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	власна система збуту	Доставляти товар до клієнтів, проводити повне впровадження системи та тестування на об'єкті	Вертикальна система	Багатоканальні системи збуту

Таблиця 6.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Власники підприємств або інвестори в галузі, що готові що впровадження новітніх технологій	Спеціалізовані форуми та виставки в галузі	Доступність та об'єктивність інформації про фірму та товар	Вплив на процес прийняття рішення про купівлю	Підкреслення переваг продукту

ВИСНОВКИ

В магістерській дисертації розроблена система автоматичного паркування автомобілів на базі технології інтернету речей. Було розглянуто компанії та технології, які вони пропонують, для рішення завдань локального позиціонування чи навігації по території паркінгів. Проведено аналіз рішень іноземних компаній FIDPARK та AFAPARK для впорядкування питань паркінгів.

Виконаний вибір технології для визначення координат мобільних об'єктів Bluetooth маяків з підтримкою протоколу Bluetooth Low Energy (LE). Стандарт що дає можливість створення Mesh-мережі. Таким чином, застосування системи локального позиціонування, побудованої на базі цих стандартів, дає змогу вирішити завдання поставлені для створення системи.

Для визначення місця розташування об'єктів обрано систему з комбінацією методів. Такий підхід дозволяє усунути недоліки одного методу, та доповнити властивостями іншого.

Розроблено структурну, функціональні схеми системи автоматичного паркування автомобілів. Для побудови інфраструктури системи локального позиціонування обрано мережеві рішення Veason-маяки.

Обрано технологію виміру відстані від мітки до базової станції та технологію передачі даних, що розроблено компанією RTL-Service. А саме за допомогою технології інерціального позиціонування IMU та комбінації методів ToF та RSSI.

Виконане моделювання траєкторії та автоматичне керування рухом авто до місця паркування у пакеті MATLAB.

Розроблено стартап-проект, у якому досліджено конкурентоспроможність системи. Розглянуті особливості системи вагомі для потенційних клієнтів та визначені стратегії розвитку та поведінки на ринку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Перехоплювальний паркінг [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%85%D0%BE%D0%BF%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%B0%D0%B2%D1%82%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%BA%D0%B0 (дата звернення 10.09.2020).
2. Транспортний колапс [Електронний ресурс] – Режим доступу до ресурсу: https://zik.ua/blogs/top_problem_kyieva_stan_sprav_ta_metody_rozviazannia_984139 (дата звернення 22.10.2020).
3. Системи керування паровкою Dahua [Електронний ресурс] – Режим доступу до ресурсу: <https://leater.com/services/sistema-upravleniya-parkovkoy.html> (дата звернення 12.09.2020).
4. AFAPARK Інтуїтивні інформаційні табло [Електронний ресурс] – Режим доступу до ресурсу: http://www.afapark.com/RU/prod_concept.html (дата звернення 12.09.2020).
5. Система контролю та оплати стоянок FIDPARK [Електронний ресурс] – Режим доступу до ресурсу: <http://fidpark.com/content/en/103/Car-Parkings-FidPark-Car.html> (дата звернення 12.09.2020).
6. Технології позиціонування в реальному часі [Електронний ресурс] – Режим доступу до ресурсу: <http://www.rtlsnet.ru/technology/view/4> (дата звернення 08.10.2020).
7. Технології локального позиціонування. Частина 1 [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/realtrac/blog/281837/> (дата звернення 09.10.2020).
8. Мережева інфраструктура системи РТЛС [Електронний ресурс] – Режим доступу до ресурсу: <http://www.rtlsnet.ru/technology/view/3> (дата звернення 09.10.2020).
9. Мережі ZigBee [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/155037/> (дата звернення 10.10.2020).

10. RFID-системи [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/194908/> (дата звернення 10.10.2018).
11. Технології позиціонування РТЛС [Електронний ресурс] – Режим доступу до ресурсу: <http://www.rtlsnet.ru/technology/view/2> (дата звернення 09.10.2020).
12. SDS-TWR вимірювання відстані [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/154445/> (дата звернення 09.10.2020).
13. Методи локального позиціонування [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/realtrac/blog/301706/> (дата звернення 15.10.2020).
14. Bluetooth mesh – базові складові мережі [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/334478/> (дата звернення 16.10.2020).
15. Bluetooth mesh мережі [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/333792/> (дата звернення 16.10.2020).
16. Bluetooth mesh – розвиток стандарту передачі даних для IoT [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/vasexperts/blog/333998/> (дата звернення 16.10.2020).
17. Інтернет речей [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82_%D1%80%D0%B5%D1%87%D0%B5%D0%B9 (дата звернення 09.10.2020).
18. Автомобіль до всього [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Vehicle-to-everything> (дата звернення 08.10.2020).
19. Виділені комунікації короткого радіусу дії [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Dedicated_short-range_communications (дата звернення 08.10.2020).
20. Автомобільні системи зв'язку [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Vehicular_communication_systems (дата звернення 08.10.2020).

21. NRF52840 Bluetooth module [Електронний ресурс] – Режим доступу до ресурсу: <http://www.skylabmodule.com/skb501/> (дата звернення 08.11.2020).
22. Products List SKYLAB [Електронний ресурс] – Режим доступу до ресурсу: <http://www.skylabmodule.com/skylab-new-arrival-uuid-major-minor-programmable-ble-smallest-ibeacon-locator-nrf52832-ble-bluetooth-4-2-motion-sensor-beacon-vg05/> (дата звернення 08.11.2020).
23. Gateway – TD05A SKYLAB [Електронний ресурс] – Режим доступу до ресурсу: <http://www.skylabmodule.com/gateway/> (дата звернення 08.11.2020).
24. GFSK – Гаусівська частотна модуляція [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/GFSK> (дата звернення 10.11.2020).
25. Модуль SPI Ethernet (ENC28J60) [Електронний ресурс] – Режим доступу до ресурсу: <http://www.kosmodrom.com.ua/pdf/ENC28J60-SPI-MOD.pdf> (дата звернення 12.11.2020).